



**Miguel Xavier dos Santos**

Licenciado em Ciências da Engenharia Electrotécnica e de Computadores

## **Sistema de Aquisição de Dados para Sinais Eletromiográficos - Implementação de um Rato para Computador**

Dissertação para obtenção do Grau de Mestre em Engenharia Electrotécnica e de Computadores

Orientador: Prof. Dr. Nuno Filipe Silva Veríssimo Paulino,  
Prof. Auxiliar, Universidade Nova de Lisboa

Júri

Presidente: Prof. Dr. Rodolfo Alexandre Duarte Oliveira - FCT/UNL

Arguente: Prof. Dr. João Pedro Abreu de Oliveira - FCT/UNL

Vogal: Prof. Dr. Nuno Filipe Silva Veríssimo Paulino - FCT/UNL



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

**Março, 2019**



## **Sistema de Aquisição de Dados para Sinais Eletromiográficos - Implementação de um Rato para Computador**

Copyright © Miguel Xavier dos Santos, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



*Para Modesto*



## Agradecimentos

Quero agradecer, em primeiro lugar, ao Professor Nuno Paulino pela total disponibilidade, apoio e supervisão dados, para este projeto, tendo-me transmitido, ao longo destes 18 meses, conhecimentos essenciais para concluir esta dissertação.

Um agradecimento muito especial à minha família - à mãe, ao pai e aos avós - pelo ânimo e apoio dados, principalmente nas etapas mais difíceis, não só deste trabalho, mas também durante todo o meu percurso académico. Por último, referir os meus amigos, por estarem sempre presentes, quando mais precisei.





## Resumo

Hoje em dia, cada vez mais o Homem beneficia da evolução tecnológica, sendo que os sistemas de aquisição de dados têm acompanhado muito essa evolução, posicionando-se na linha da frente para contribuir para a melhoria de diversas áreas da sociedade, com especial ênfase na saúde. Esta dissertação tem por objetivo desenvolver um sistema de aquisição de dados para sinais eletromiográficos, com vista à implementação de um rato para computador, controlado por este mesmo tipo de sinais biomédicos.

Primeiramente, fez-se uma revisão bibliográfica de métodos, técnicas e de projetos já desenvolvidos, com características semelhantes ao trabalho aqui proposto. Seguiu-se o escrutínio, das características dos sinais musculares e, posteriormente, uma análise dos vários processos de condicionamento, processamento e classificação do sinal, bem como modelos de aplicação existentes.

Os conhecimentos adquiridos permitiram, depois, realizar uma fase de testes, com análise de resultados, em que se desenvolveram vários ensaios distintos, no que diz respeito à captação do sinal, bem como ao circuito de amplificação e ao processamento realizado, quer em Matlab, quer em PSoC.

A última etapa diz respeito à implementação do rato para computador, apresentando, essencialmente, a montagem realizada no PSoC e os passos referentes à programação e à configuração das várias componentes utilizadas, para comandar este periférico através de sinais musculares.

**Palavras-chave:** Sistemas de Aquisição de Dados, Sinais Eletromiográficos, Condicionamento do Sinal, Processamento do Sinal, PSoC.



# Abstract

Nowadays, Man is being benefited from technological evolution, so that data acquisition systems have been increasingly developed, helping many society's areas, such as the health. This dissertation seeks to develop a data acquisition system for electromyographic signals, looking for the implementation of a computer mouse, controlled by this type of biomedical signals.

First of all, a bibliographical review of methods, techniques and projects, similar to the work to be developed, was done. We began by scrutinizing the characteristics of the muscular signals, followed by an analysis of several processes concerning signal's conditioning, processing and classification, as well as, existing application models.

The acquired knowledge allowed us to carry out a testing phase followed by results analysis, in which several different tests were developed, mainly regarding the signal acquisition, but also, with great emphasis on the amplification circuit and the processing performed in either Matlab or PSoC.

The last step concerns the mouse's implementation, and presents essentially the assembly performed in the PSoC, and the steps regarding programming and configuration of the various components used to control this peripheral, through muscle signals.

**Keywords:** Data Acquisition System, Electromyographic Signals, Signal Conditioning, Signal Processing, PSoC.



# Índice

Índice de Figuras .....	xv
Índice de Tabelas.....	xvii
Acrónimos .....	xix
Capítulo 1 Introdução.....	1
Capítulo 2 Estado da Arte .....	3
2.1.    Sinais Biomédicos .....	3
2.1.1.    Eletrocardiograma (ECG).....	3
2.1.2.    Eletroencefalograma (EEG) .....	4
2.1.3.    Eletromiograma (EMG).....	5
2.2.    Eléttodos.....	7
2.3.    Condicionamento do Sinal Eletromiográfico .....	10
2.3.1.    Amplificadores e Filtros .....	10
2.3.2.    Circuitos e Modelos para Condicionamento do Sinal .....	12
2.4.    Processamento do Sinal Eletromiográfico.....	16
2.4.1.    Conversor Analógico-Digital .....	16
2.4.2.    Dispositivos de Aquisição de Dados .....	17
2.5.    Classificação do Sinal Eletromiográfico .....	19
2.5.1.    Rede Neuronal Artificial .....	20
2.6.    Modelos de Aplicação .....	23
Capítulo 3 Testes e Análise de Resultados.....	25
3.1.    Fase 1 – Captação do Sinal.....	25
3.2.    Fase 2 – Circuito de Amplificação .....	30
3.3.    Fase 3 – Remoção de <i>Offset</i> .....	35
3.4.    Fase 4 – Níveis de Força .....	37
3.5.    Fase 5 – Processamento do Sinal com PSoC 5LP .....	39
3.6.    Fase 6 – Processamento do Sinal com Matlab .....	42
Capítulo 4 Implementação Final .....	47
Capítulo 5 Conclusões e Trabalho Futuro .....	61

5.1. Conclusões .....	61
5.2. Trabalho Futuro.....	63
Bibliografia.....	65
Apêndices .....	67

## Índice de Figuras

Figura 1.1: Sistema de aquisição de dados (Adaptado de [1]).	2
Figura 2.1: ECG de individuo do sexo masculino (Adaptado de [2]).	4
Figura 2.2: a) ritmo delta; b) ritmo teta; c) ritmo alfa; d) ritmo beta; (Adaptado de [2]).	5
Figura 2.3: Potencial de ação (Adaptado de [2]).	6
Figura 2.4: Eletromiograma do braço (Adaptado de [2]).	6
Figura 2.5: Eléctrodo intramuscular (Adaptado de [6]).	8
Figura 2.6: Eléctrodo de superfície.	9
Figura 2.7: Colocação do eléctrodo (Adaptado de [5]).	9
Figura 2.8: Amplificador de instrumentação (Adaptado de [7]).	10
Figura 2.9: Filtro passa-baixo de primeira ordem passivo (Adaptado de [8]).	11
Figura 2.10: Filtro passa-alto de primeira ordem passivo (Adaptado de [9]).	12
Figura 2.11: Exemplo 1 de modelo de amplificação biomédico (Adaptado de [10]).	13
Figura 2.12: Exemplo 1 do sinal amplificado (Adaptado de [10]).	14
Figura 2.13: Exemplo 2 de modelo de amplificação biomédico (Adaptado de [11]).	14
Figura 2.14: Exemplo 2 do sinal amplificado (Adaptado de [11]).	15
Figura 2.15: ADC (Adaptado de [12]).	16
Figura 2.16: Placa de som (Adaptado de [5]).	17
Figura 2.17: <i>Simulink</i> para placa de som (Adaptado de [5]).	18
Figura 2.18: Sinal filtrado (Adaptado de [10]).	18
Figura 2.19: Sinal retificado (Adaptado de [10]).	18
Figura 2.20: <i>Single layer perceptron</i> (Adaptado de [13]).	20
Figura 2.21: <i>Multi-layer perceptron</i> (Adaptado de [13]).	21
Figura 2.22: Classificação de sinais EMG para o movimento da mão (Adaptado de [14]).	22
Figura 2.23: Mão robótica (Adaptado de [16]).	24
Figura 3.1: Eléctrodo de superfície Lifecare – 40X40MM.	26
Figura 3.2: Eléctrodo de superfície Lifecare - 45MM.	26
Figura 3.3: Dispositivo de aquisição de dados NI myDAQ - National Instruments (Adaptado de [17]).	27
Figura 3.4: Inputs e outputs do dispositivo de aquisição de dados NI myDAQ (Adaptado de [17]).	27
Figura 3.5: Bíceps relaxado – Fase 1.	28
Figura 3.6: Bíceps contraído – Fase 1.	28
Figura 3.7: Frontal relaxado – Fase 1.	29
Figura 3.8: Frontal contraído – Fase 1.	30
Figura 3.9: Placa de amplificação para sinais biomédicos.	31
Figura 3.10: Diagrama de conexões do AD620 (Adaptado de [18]).	31
Figura 3.11: Esquema simplificado do AD620 (Adaptado de [18]).	32
Figura 3.12: Filtragem dos sinais de entrada.	32
Figura 3.13: Circuito para a tensão de referência.	33
Figura 3.14: Bíceps relaxado – Fase 2.	34

Figura 3.15: Bíceps contraído – Fase 2. ....	34
Figura 3.16: Flexor radial do carpo do braço direito – Fase 3 (Adaptado de [19]). ....	35
Figura 3.17: Flexor radial do carpo relaxado – Fase 3. ....	36
Figura 3.18: Flexor radial do carpo contraído – Fase 3. ....	37
Figura 3.19: Contração nível 1.....	38
Figura 3.20: Contração nível 2.....	38
Figura 3.21: Contração nível 3.....	39
Figura 3.22: CY8CKIT-059 PSoC 5LP Prototyping Kit (Adaptado de [20]). ....	40
Figura 3.23: Delta_Sigma ADC – PSoC 5LP. ....	40
Figura 3.24: Esquemático para digitalização do sinal – PSoC 5LP. ....	41
Figura 3.25: Contrações musculares digitalizadas – Bridge Control Panel. ....	42
Figura 3.26: Contração muscular digitalizada– Bridge Control Panel. ....	42
Figura 3.27: Contração muscular digitalizada – Matlab.....	43
Figura 3.28: Espectro de frequências com contração muscular – Matlab.....	44
Figura 3.29: Espectro de frequências sem contração muscular – Matlab.....	44
Figura 3.30: Resposta em frequência do filtro FIR – Matlab.....	45
Figura 3.31: Contração muscular digitalizada com efeito do FIR– Matlab. ....	46
Figura 3.32: Contração muscular digitalizada com filtro (azul), sem filtro (vermelho).....	46
Figura 4.1: Comando: a) direita; b) baixo; c) cima; d) esquerda. ....	48
Figura 4.2: Esquemático da implementação final. ....	49
Figura 4.3: Parâmetros do Delta_Sigma ADC. ....	49
Figura 4.4: Parâmetros do Filtro Passa-Banda. ....	51
Figura 4.5: Eletromiograma do Comando Esquerda. ....	54
Figura 4.6: Eletromiograma do Comando Direita. ....	55
Figura 4.7: Eletromiograma do Comando Baixo. ....	56
Figura 4.8: Eletromiograma do Comando Cima. ....	56
Figura 4.9: Eletromiograma Sem Comando. ....	57
Figura 4.10: Detetor de pico (Adaptado de [21]). ....	57
Figura 4.11: Sinusoide e detetor de pico (Adaptado de [22]). ....	58
Figura 4.12: Contração do Flexor - Detetor de Pico. ....	59
Figura 4.13: Evolução do sinal desde a captação ao processamento - a) Sinal sem amplificação; b) Sinal amplificado; c) Sinal filtrado; d) Sinal após detetor de pico. ....	60
Figura 5.1: Montagem da Implementação Final. ....	63



## Índice de Tabelas

Tabela 2.1: Estatísticas do sinal mioelétrico (Adaptado de [10]).	19
---	----



## Acrónimos

<b>ACK</b>	<i>Acknowledge</i>
<b>ADC</b>	<i>Analog to Digital Converter</i>
<b>AI</b>	<i>Analog Input</i>
<b>AMPOP</b>	<i>Amplificador Operacional</i>
<b>AVC</b>	<i>Acidente Vascular Cerebral</i>
<b>DAC</b>	<i>Digital to Analog Converter</i>
<b>DAQ</b>	<i>Data Acquisition System</i>
<b>DC</b>	<i>Direct Current</i>
<b>DMA</b>	<i>Direct Memory Access</i>
<b>COM</b>	<i>Communication Port</i>
<b>ECG</b>	<i>Eletrocardiograma</i>
<b>EEG</b>	<i>Eletroencefalograma</i>
<b>EMG</b>	<i>Eletromiograma</i>
<b>eoc</b>	<i>end of conversion</i>
<b>fft</b>	<i>fast fourier transform</i>
<b>HID</b>	<i>Human Interface Device</i>
<b>MAV</b>	<i>Mean Absolute Value</i>
<b>Med</b>	<i>Median Value</i>
<b>MLP</b>	<i>Multi-Layer Perceptron</i>
<b>PSoC</b>	<i>Programmable system-on-chip</i>
<b>RNA</b>	<i>Rede Neuronal Artificial</i>
<b>SLP</b>	<i>Single Layer Perceptron</i>
<b>SPS</b>	<i>Samples Per Second</i>
<b>UART</b>	<i>Universal Asynchronous Receiver-Transmitter</i>
<b>USB</b>	<i>Universal Serial Bus</i>

**USBFS** *Universal Serial Bus Full Speed*

**VAR** *Variance*

**WL** *Waveform Length*



# 1

## Introdução

Nos pontos seguintes, pretende-se contextualizar e fundamentar a escolha do sistema de aquisição de dados, para a recolha de sinais biomédicos, fazendo a ligação do mesmo com um rato para computador, que será controlado pelos referidos sinais registados. Em primeiro lugar, far-se-á uma exposição das motivações para a elaboração deste projeto bem como dos principais objetivos a atingir durante o trabalho.

Nos dias de hoje, cada vez mais o Homem tem necessidade de recorrer a tecnologias que incidam na aquisição de dados. Em concreto, o avanço tecnológico registado nos últimos anos e o sucesso de muitos projetos resultam, em grande parte, da aquisição e tratamento desses dados. É, por isso mesmo, uma área científica com grande destaque na atualidade e perspetiva-se que venha a ter um grande impacto no futuro da humanidade.

A aquisição de dados suporta diversos campos da ciência, tendo na sua base a recolha e o processamento de sinais, extraindo os dados dos mesmos. Estes dados são posteriormente trabalhados e analisados, para serem aplicados de diversas formas, consoante a área de estudo em questão.

A Figura 1.1 apresenta um esquema geral de um sistema de aquisição de dados. Analisando o esquemático, observa-se que o sensor deteta a ocorrência de um fenómeno físico, levando a que o sinal detetado seja, de seguida, condicionado<sup>1</sup>. O passo seguinte, remete para a utilização de um *hardware*, que permita a aquisição do sinal previamente condicionado e que contenha um ADC (*Analog to Digital Converter*). Ou seja, um conversor capaz de converter o sinal analógico, proveniente do fenómeno físico, num sinal digital que é, posteriormente, estudado com recurso a um computador e a um *software* específico para este efeito. O processo pode terminar neste ponto, com o utilizador a visualizar os dados

---

<sup>1</sup> As técnicas mais comuns para o condicionamento do sinal passam pela amplificação, com um amplificador, e pela filtragem, com um filtro.

recebidos no computador, ou pode alimentar um desenvolvimento maior ou ainda pode servir para desencadear um novo processo, com a recolha original de dados a suportar desenvolvimentos com finalidades diferentes. Em qualquer caso, é necessário recorrer a um atuador<sup>2</sup>, que terá a capacidade de distinguir o que fazer, quando recebe um determinado comando proveniente do computador.

Como é conhecido, a área da saúde, com as suas várias especialidades, é cada vez mais um campo que tem vindo a beneficiar muito da evolução tecnológica, pelo que, depois de alguma pesquisa, concluiu-se que seria interessante explorar a aquisição de dados através de sinais musculares e demonstrar a pertinência da conversão destes sinais em comandos para o desempenho de determinadas tarefas. Considerando o caso de pessoas com deficiência, é possível perceber que estes tipos de técnicas são ideais para lhes proporcionar uma melhor qualidade de vida. Veja-se, por exemplo, a situação de uma pessoa que não consegue mexer ambas as mãos, na sequência de algum tipo de acidente ou doença, ou até mesmo de uma pessoa que tenha sofrido uma amputação. Nestes casos, verificam-se limitações severas que impedem o indivíduo em causa de desempenhar algumas tarefas básicas, com a prontidão que pretendia.

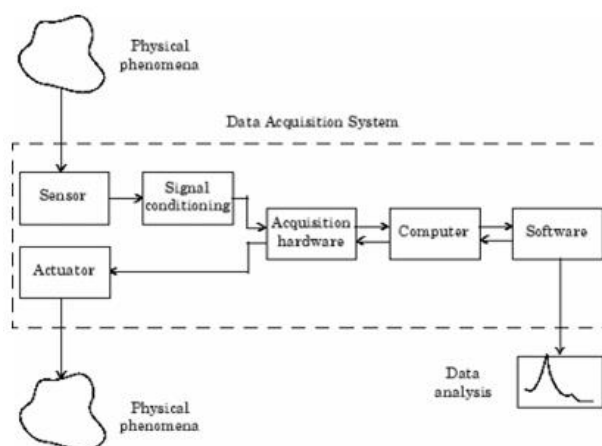


Figura 1.1: Sistema de aquisição de dados (Adaptado de [1]).

Tendo isto em conta e o facto de o computador ter passado a fazer parte do quotidiano do Homem, é de grande relevância encontrar alternativas, que permitam ajudar a melhorar a qualidade de vida das pessoas nestas situações, até no sentido de facilitar a sua integração no mercado de trabalho.

Como foi mencionado no início deste capítulo, o objetivo deste projeto passa pela criação de um rato para computador, com recurso a sinais biomédicos. Obviamente, que existem diversos tipos de sinais biomédicos, mas, neste caso, o que vamos analisar são os sinais dos músculos. Concretamente, será através dos impulsos elétricos que irão ocorrer, aquando da contração de um músculo, que o utilizador poderá controlar o rato.

<sup>2</sup> Um atuador é a parte de uma máquina responsável pelo movimento e controlo de um mecanismo ou sistema.

# 2

## Estado da Arte

### 2.1. Sinais Biomédicos

A base desta dissertação está nos sinais biomédicos, mais em concreto nos sinais musculares. No entanto, é importante perceber por que motivo se optou por este tipo de sinais, em detrimento de outros, fazendo uma descrição detalhada dos mesmos, no que concerne à análise da onda produzida para os diversos tipos de sinais existentes. Sabe-se, também, que o corpo humano produz vários tipos de sinais diferentes: bioquímicos, físicos e elétricos. Para o caso em estudo, interessa analisar as ondas elétricas, que se manifestam na forma de potencial ou de corrente, para que o sistema funcione.

De seguida, far-se-á uma exposição de três tipos principais de sinais elétricos que o corpo humano emite.

#### 2.1.1. Eletrocardiograma (ECG)

O primeiro caso de estudo trata-se do eletrocardiograma (ECG). De acordo com [2] um ECG é a manifestação elétrica da contração do músculo cardíaco e, para além de ser o mais conhecido, é também, normalmente, o sinal biomédico mais usado. Tomando em conta a Figura 2.1, trata-se de analisar os vários impulsos do batimento cardíaco de um indivíduo do sexo masculino com 24 anos.

Da observação, verifica-se que há um batimento cardíaco uniforme ao longo do intervalo de tempo, sendo de destacar três picos de maior amplitude, perto de 1mV com um intervalo de duração bastante curto, cada um deles. Desta forma, é possível entender que não haverá nunca grande discrepância no sinal cardíaco detetado para este indivíduo, como para qualquer outro, a não ser nos casos de doença cardíaca, em que o ritmo cardíaco pode surgir significativamente alterado.



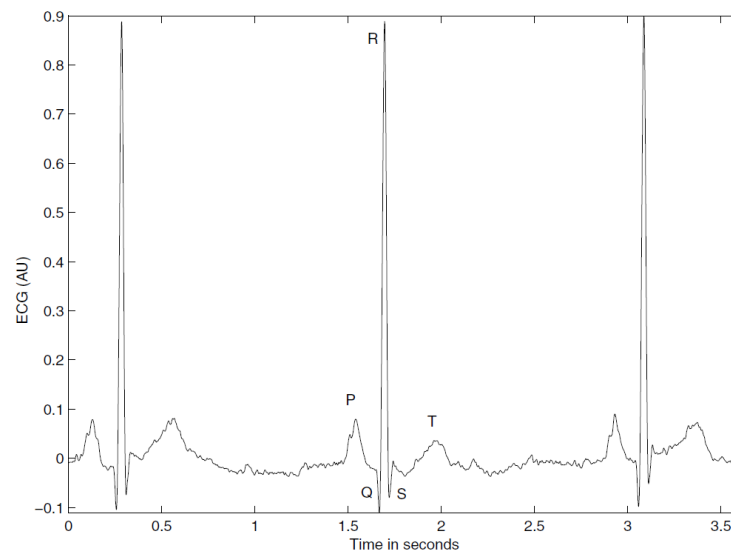


Figura 2.1: ECG de individuo do sexo masculino (Adaptado de [2]).

Perante o padrão de regularidade desta recolha e, sobretudo, pela impossibilidade do utilizador poder controlar estes batimentos, percebe-se que o eletrocardiograma não é o sinal adequado para o tipo de projeto que se pretende desenvolver.

### 2.1.2. Eletroencefalograma (EEG)

Outro caso de estudo é o eletroencefalograma (EEG). Pela referência [2], sabe-se que um EEG representa a atividade elétrica do cérebro que, ao contrário dos impulsos cardíacos, pode variar bastante, visto que a complexidade e diversidade da atividade do cérebro humano leva a que esteja permanentemente a emitir ondas com diferentes amplitudes e frequências. Constitui, por isso, uma muito interessante e complexa fonte de pesquisa, entre outros aspetos, para tentar perceber quais as ondas que são mais comuns e o que as desencadeia bem como que sinais resultam de pensamentos/comandos e que áreas do cérebro os produzem.

Em primeiro lugar, verifica-se que, em regra, uma onda típica do cérebro tem uma amplitude que varia entre  $10 \mu\text{V}$  e  $100 \mu\text{V}$ , sendo de destacar as várias gamas de frequências distintas, consoante o tipo de onda:

- Delta ( $\delta$ ):  $0.5 \leq f < 4 \text{ Hz}$ ;
- Teta ( $\theta$ ):  $4 \leq f < 8 \text{ Hz}$ ;
- Alfa ( $\alpha$ ):  $8 \leq f \leq 13 \text{ Hz}$ ;
- Beta ( $\beta$ ):  $f > 13 \text{ Hz}$ .

A Figura 2.2 ilustra cada um deste tipo de ondas. A onda a) é referente, ao tipo Delta e regista-se principalmente em adultos a dormir e em recém-nascidos, sendo a onda b) do tipo Teta, que surge em crianças e em adultos com sonolência. Os sinais, de maior frequência, c) e d) são respetivamente do tipo Alfa e Beta, surgindo o primeiro quando o indivíduo está relaxado e o segundo em momentos de grande ansiedade ou, então, num momento de maior ênfase em termos de pensamento ou raciocínio.

Dado que se verifica uma disparidade significativa nas amplitudes e frequências das ondas cerebrais, este método seria bastante interessante de desenvolver e trabalhar. Porém é necessário ter em conta que, como foi mencionado anteriormente, se está perante amplitudes de sinal na ordem dos  $\mu V$ , o que tornaria bastante complexo o seu processo de amplificação, pelo que acaba por não ser a solução ideal para este projeto.

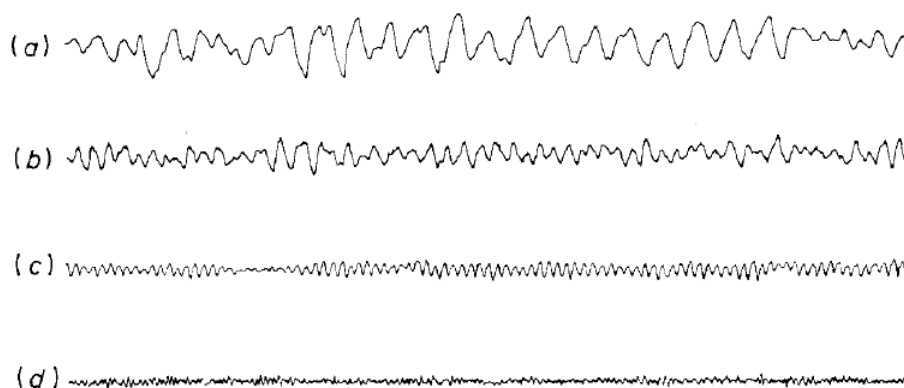


Figura 2.2: a) ritmo delta; b) ritmo teta; c) ritmo alfa; d) ritmo beta; (Adaptado de [2]).

### 2.1.3. Eletromiograma (EMG)

O eletromiograma, EMG, traduz o sinal elétrico, proveniente de um determinado músculo, aquando da sua contração. Por isso e conforme já descrito, constitui a base deste trabalho, sendo necessário analisar e descrever este sinal biomédico de forma mais detalhada, em relação aos anteriores.

Em concreto, quando os músculos de uma pessoa estão relaxados, i.e., sem qualquer atividade relevante, a onda elétrica produzida é praticamente insignificante, o que implica que não haja produção de corrente elétrica, situação denominada de não existência de potencial de ação<sup>3</sup>. Em contrapartida, a movimentação muscular, que se traduz na sua contração, desencadeia grandes variações de amplitude no eletromiograma. Ou seja, surgem os potenciais de ação nas células, como se ilustra na Figura 2.3.

---

<sup>3</sup> A referência [2] explica que o potencial de ação é o componente básico de qualquer sinal bioelétrico, garantido informação em termos fisiológicos da atividade ao nível de cada célula. Assim, manifesta-se na forma de um sinal elétrico, acompanhando a contração mecânica de cada célula muscular, quando estimulada por uma determinada corrente elétrica (externa ou proveniente de neurónios).

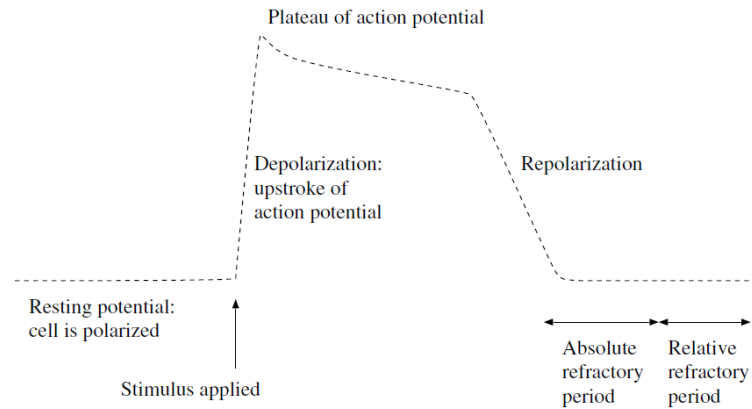


Figura 2.3: Potencial de ação (Adaptado de [2]).

Segundo o que foi indicado na nota anterior de rodapé, este sinal evolui de uma situação em que a célula se encontra em repouso e sofre um determinado estímulo, que leva a um pico de amplitude na ordem dos mV, para um novo estado de repouso, necessário, e em que não poderá existir novo potencial de ação (*absolute refractory period*). É de referir, ainda, que todo este processo se desencadeia numa unidade motora, ou seja, o neurónio motor produz o potencial de ação que será transmitido às fibras musculares a que está ligado e a transmissão de diversos sinais elétricos para um determinado músculo, leva a uma sobreposição dos mesmos, o que origina propriamente o EMG, como se constata pela Figura 2.4.

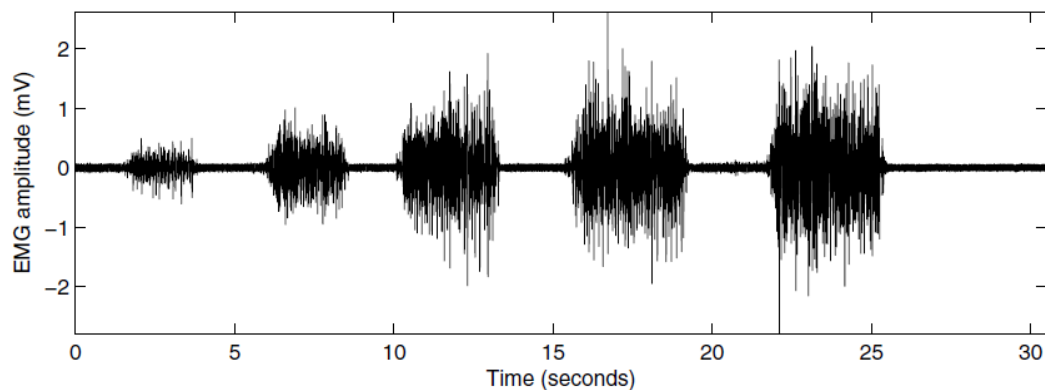


Figura 2.4: Eletromiograma do braço (Adaptado de [2]).

Sabendo que o sinal apresentado se trata de um eletromiograma, registado num braço de um determinado indivíduo, as características do mesmo são as seguintes: no momento em que o músculo está em repouso, não se regista praticamente qualquer oscilação de amplitude, visto que não há transmissão de potenciais de ação provenientes da célula nervosa. De seguida, há uma primeira flexão

do músculo e verificam-se vários picos negativos e positivos, na ordem dos mV, já que os potenciais de ação ocorrem, segundo a referência [3], em intervalos de tempo aleatórios.

Ao longo do tempo, o processo é todo ele igual: as células musculares entram novamente em repouso, sendo de seguida excitadas e assim sucessivamente. É importante salientar, como elementos diferenciadores, o aumento ou diminuição das amplitudes de tensão em valor absoluto, o que significa que o indivíduo contraiu com mais ou menos força o seu braço, e a duração com que o faz. Se contrair durante mais tempo, as perturbações no sinal têm maior duração na escala de tempo.

Por tudo o que foi mencionado nesta secção, mas também nas duas anteriores, entende-se que a recolha de sinais derivados de contração muscular será a melhor opção a desenvolver, sobretudo porque os sinais são bastantes distintos uns dos outros e, ao mesmo tempo, facilmente controláveis pela pessoa. Por outro lado e como referido, a indispensável diversidade destes sinais é relativamente fácil de obter, fazendo variar a força, a cadência e a duração da contração dos músculos.

Mais à frente, será abordada a questão dos níveis de força, sendo que é desejável que os valores de amplitude das ondas sejam suficientemente significativos, para serem facilmente amplificados e posteriormente analisados.

## 2.2. Eléktrods

Tendo-se optado pelo eletromiograma, é necessário decidir qual o melhor equipamento a utilizar, para conseguir efetuar uma eletromiografia<sup>4</sup>, com eficácia. O primeiro passo, diz respeito à escolha dos eléctrods para a deteção do sinal.

Em concreto, existem dois tipos de eletromiografia, a intramuscular e a superficial. No caso da primeira, são utilizados eléctrods de perfuração, com formato de agulha (Figura 2.5) e, normalmente, o objetivo passa por registar o sinal elétrico dos músculos mais internos. A segunda, utiliza eléctrods de superfície (Figura 2.6) e destina-se a músculos mais superficiais.

Como se trata de um projeto académico e visto não existirem competências médicas, os únicos eléctrods disponíveis serão os de superfície, já que não trarão qualquer lesão ao utilizador do equipamento, permitindo um processo de tentativa/erro na colocação do eléctrodo, de forma a detetar o melhor possível o sinal que se pretende. Esta questão é bastante relevante, tal como indica a referência [4], dado que os valores de amplitude do sinal mioelétrico são tanto maiores, quanto menor for a distância entre o eléctrodo e a fibra muscular. Para além disso, seguindo a informação de [5], o sinal que se consegue detetar raramente é “limpo”. Ou seja, há sempre diversos tipos de interferência, o chamado ruído, provenientes muitas vezes do meio circundante, tais como:

---

<sup>4</sup> A eletromiografia é a técnica que permite detetar um eletromiograma.

- Radiação eletromagnética - a magnitude deste tipo de ruído ao nível da pele pode ser até três vezes superior ao EMG. Para além disto, qualquer instrumento eletromagnético gera este tipo de radiação contribuindo para o aumento deste ruído, levando a que seja bastante complicado evitá-lo;
- Imperfeições eletrónicas - os componentes eletrónicos usados para a deteção do sinal, não só os elétrodo, têm ruído inerente, i.e., todos geram ruído elétrico. Neste caso, não é possível eliminá-lo, mas é possível reduzi-lo, com recurso a equipamentos eletrónicos de qualidade elevada;
- Ruído *baseline* - surge no contacto dos elétrodo com a pele, pelo que é de grande relevância preparar a mesma antes da colocação do elétrodo. A magnitude deste ruído é proporcional à resistência da superfície do elétrodo. Logo, aumentar a área do mesmo e limpá-la faz com que o ruído seja reduzido mas não é possível eliminar o mesmo;
- Ruído fisiológico - por vezes, pode ocorrer interferência de outros sinais biomédicos, por exemplo ECG, contribuindo assim como ruído. Também não é possível eliminá-lo, mas é possível minimizar o efeito, através da colocação correta dos elétrodo;
- Ruído de alimentação – o equipamento de deteção tem de ser alimentado, mas com recurso a baterias, de modo a minimizar a perturbação do sinal mioelétrico causada pela tensão das tomadas (230V-50Hz). O ideal é utilizar este tipo de equipamentos numa zona isolada, sem influência da corrente elétrica;
- Ruído nos cabos elétricos - Os cabos que fazem a transmissão do sinal entre os elétrodo e a placa de aquisição de dados, também são municiadores de algum ruído.



Figura 2.5: Elétrodo intramuscular (Adaptado de [6]).

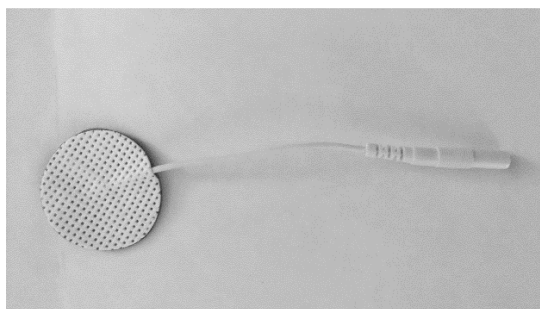


Figura 2.6: Eléctrodo de superfície.

Por último, mas não menos importante, pode verificar-se o efeito de *cross-talk*. Não sendo propriamente um tipo de ruído, acaba por ter o mesmo efeito, visto que se trata do conjunto de sinais emitidos pelos músculos circundantes relativamente ao músculo do qual pretendemos detetar o EMG. A melhor solução passa por utilizar eléctrodos de dimensões reduzidas e diminuir o espaçamento entre cada eléctrodo, focando assim a deteção do sinal numa área a mais reduzida possível.

Obviamente que, quanto melhor for a colocação do sensor<sup>5</sup>, melhor será o sinal que se obtém, pelo que o ideal é mesmo colocá-lo no centro do músculo e o mais distante possível de tendões e zonas de inervação<sup>6</sup>, como se verifica na Figura 2.7. Isto facilmente se justifica, pois as fibras musculares no centro dos músculos têm maior diâmetro do que junto aos tendões e, sendo o potencial de ação diretamente proporcional ao diâmetro das fibras, necessariamente o EMG atingirá amplitudes bem mais significativas, nessa zona. Por outro lado, a colocação de um eléctrodo numa zona de inervação, muitas vezes levaria à deteção do cancelamento dos potenciais de ação, levando o sensor a registar amplitudes bastante mais reduzidas.

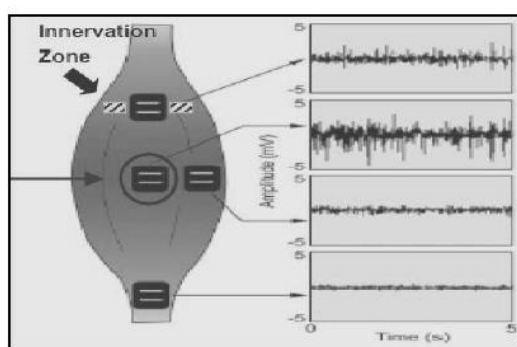


Figura 2.7: Colocação do eléctrodo (Adaptado de [5]).

<sup>5</sup> Neste caso, para este projeto, os eléctrodos funcionam como sensores, visto que detetam o sinal.

<sup>6</sup> As zonas de inervação dizem respeito a zonas com presença acentuada de nervos.

## 2.3. Condicionamento do Sinal Eletromiográfico

Esta secção pretende expor vários métodos possíveis, para o condicionamento do sinal mioelétrico. Tal como foi mencionado anteriormente, o condicionamento do sinal é essencial neste projeto, para ultrapassar os constrangimentos criados pelas baixas amplitudes deste tipo de onda biomédica e pelo ruído que está permanentemente associado ao sinal. Daí que seja necessário recorrer a amplificadores e filtros, de forma a solucionar estas questões.

### 2.3.1. Amplificadores e Filtros

#### 2.3.1.1. Amplificador de Instrumentação

Em termos de amplificadores, o mais comum é usar-se montagens com um tipo particular de amplificadores diferenciais<sup>7</sup>, os amplificadores de instrumentação. Um exemplo típico do circuito de um amplificador deste género pode ser visto, na Figura 2.8.

É importante descrever o funcionamento deste circuito, pois serve como base para a grande maioria dos amplificadores de instrumentação existentes que, como referido, são os mais adequados para amplificar sinais biomédicos. Segundo a referência [7], o esquema inclui dois *buffers*<sup>8</sup> à entrada do circuito com três resistências a ligá-los: duas com o mesmo valor, nos extremos, e a do meio,  $R_{\text{gain}}$ , com um valor diferente, sendo que o *feedback* negativo permite que nos pontos 1 e 2 a tensão seja exatamente a mesma que está imposta nos pontos  $V_1$  e  $V_2$ , respetivamente. Na saída, o circuito tem um amplificador operacional.

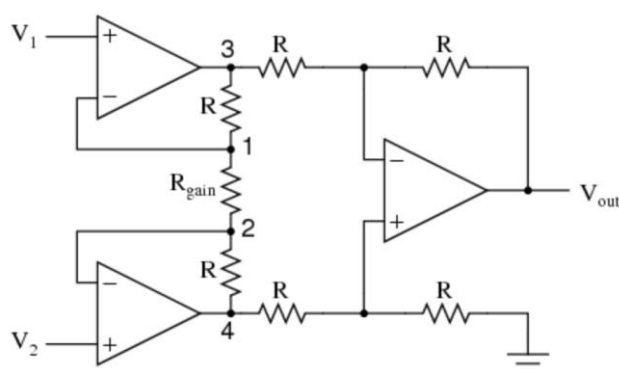


Figura 2.8: Amplificador de instrumentação (Adaptado de [7]).

<sup>7</sup> Um amplificador diferencial tem a capacidade de amplificar a diferença entre duas tensões de entrada.

<sup>8</sup> Um *buffer* é um seguidor de tensão, ou seja, um amplificador operacional com ganho de tensão de 1.

A queda de tensão em  $R_{gain}$ , que é a mesma entre  $V_1$  e  $V_2$ , implica que  $R_{gain}$  seja “percorrida” por uma corrente, que é a mesma para as referidas 3 resistências, já que não há perda de corrente em cada um dos *buffers*. Logo, é possível definir a tensão entre os pontos 3 e 4 pela expressão (2.1):

$$V_{3,4} = (V_2 - V_1) \times \left(1 + \frac{2R}{R_{gain}}\right) \quad (2.1)$$

Tendo agora em consideração o AMPOP de saída do circuito, este amplifica a tensão de (2.1) com um ganho de 1. Todas as 6 resistências  $R$  têm o mesmo valor, pelo que o ganho fica definido pela equação (2.2):

$$A_v = \frac{V_{out}}{(V_2 - V_1)} = \left(1 + \frac{2R}{R_{gain}}\right) \times \frac{R}{R} = \left(1 + \frac{2R}{R_{gain}}\right) \quad (2.2)$$

É ainda importante realçar, que o ganho mais baixo possível deste circuito seria mesmo 1 e aconteceria no caso hipotético em que o valor de  $R_{gain}$  tendesse para infinito, ou seja circuito aberto.

#### 2.3.1.2. Filtro Passa-Baixo

Em relação a filtros, verifica-se que há diversas abordagens distintas, sendo que um dos mais utilizados é o filtro passa-baixo. Tal como o nome indica, este tipo de filtro deixa passar os sinais de frequências mais baixas, entre os 0 Hz e a frequência de corte estabelecida, eliminando as frequências acima desse limiar.

Existem diversas abordagens possíveis, para este tipo de componente eletrónica: soluções mais simples ou mais complexas. Como exemplo, apresenta-se o filtro passa-baixo de primeira ordem passivo<sup>9</sup>, representado na Figura 2.9.

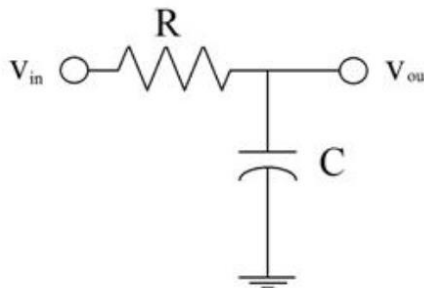


Figura 2.9: Filtro passa-baixo de primeira ordem passivo (Adaptado de [8]).

<sup>9</sup> Um filtro passivo utiliza exclusivamente elementos passivos, como bobines, resistências e condensadores.



Trata-se de um circuito bastante simples, sendo composto apenas por uma resistência e um condensador. Tendo em conta que o condensador oferece grande resistência a sinais de baixas frequências e que, pelo contrário, é bastante permeável a sinais de altas frequências, este circuito tem o comportamento desejado e que consiste em deixar que apenas as baixas frequências surjam na saída  $V_{out}$ . Neste caso, para as frequências mais altas, o condensador à massa atua como um curto-circuito. Refira-se, ainda, que a frequência de corte para este circuito é dada pela expressão (2.3):

$$f_c = \frac{1}{2\pi RC} \quad (2.3)$$

### 2.3.1.3. Filtro Passa-Alto

Apresenta-se, de seguida, o filtro com a função exatamente oposta, face ao anterior. Agora o objetivo passa por filtrar as altas frequências, portanto o circuito deve deixar passar as frequências acima da frequência de corte, eliminando as frequências mais baixas. Novamente, são possíveis diversas implementações, sendo dado como exemplo o filtro passa-alto de primeira ordem passivo, representado na Figura 2.10.

Em termos de elementos, o circuito em causa, é idêntico ao da secção anterior, sendo a inversão funcional resultante da troca de posições entre o condensador e a resistência. Tal como já foi mencionado, o condensador à entrada é como um circuito aberto para os sinais de baixas frequências e é como um curto-circuito para altas frequências que passam para a saída  $V_{out}$ . A equação da frequência de corte é exatamente a mesma do filtro passa-baixo (expressão (2.3)).

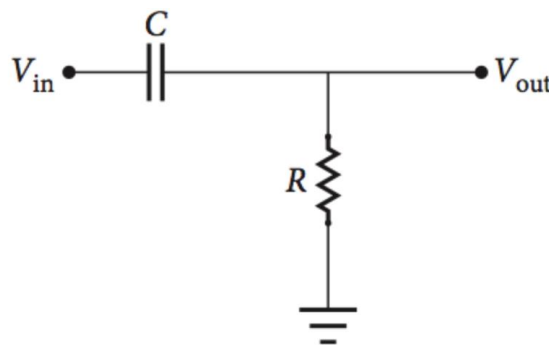


Figura 2.10: Filtro passa-alto de primeira ordem passivo (Adaptado de [9]).

### 2.3.2. Circuitos e Modelos para Condicionamento do Sinal

Concluída a apresentação dos circuitos elementares - amplificador de instrumentação e as tipologias mais comuns em termos de filtros -, importa agora expor algumas das principais montagens existentes,

com recurso a este tipo de elementos e que são adequados a amplificar e filtrar sinais biomédicos, particularmente eletromiogramas.

### 2.3.2.1. Modelo 1

O primeiro caso que se pretende abordar, diz respeito à montagem da Figura 2.11.

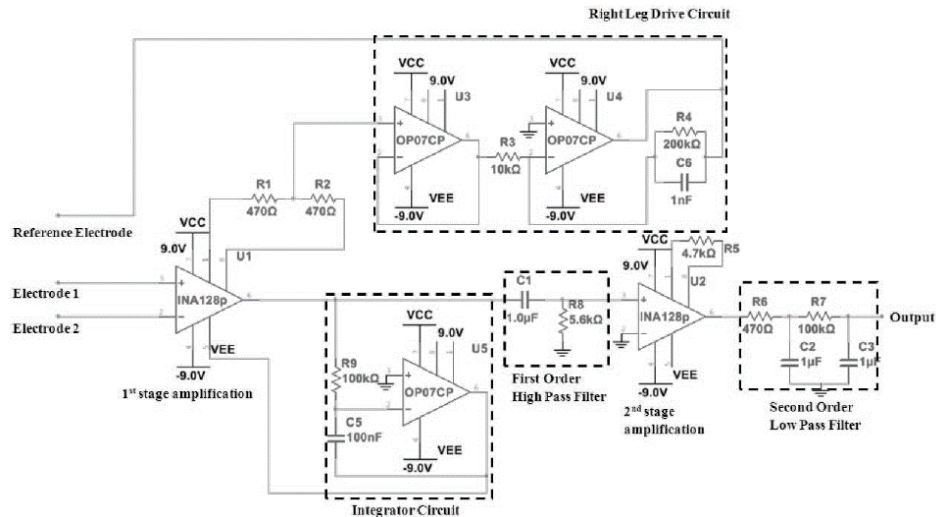


Figura 2.11: Exemplo 1 de modelo de amplificação biomédico (Adaptado de [10]).

A referência [10], indica-nos que a amplitude de um sinal mioelétrico varia entre 1 e 10 mV, tendo sido implementado o circuito da Figura 2.11 para amplificar a onda detetada pelos dois eléttrodo. Há ainda um terceiro eléttrodo que serve de referência. Nesta montagem estão presentes os elementos já abordados anteriormente e também outros com relevância para a eficácia do funcionamento deste sistema.

Primeiramente, salienta-se o bloco *right leg drive circuit*, com a função de reduzir o ganho de modo comum<sup>10</sup>. Portanto, o sinal é detetado pelos eléttrodo 1 e 2 entra num primeiro bloco de amplificação, utilizando o amplificador de instrumentação INA128. O sinal à saída deste bloco passa por um circuito de integração, com uma  $f_c = 15,91$  Hz, para remover a tensão DC de *offset*, caso esta exista.

Por fim, tem-se um filtro passa-alto de primeira ordem ( $f_c = 28$  Hz) e um filtro passa-baixo de segunda ordem ( $f_c = 742$  Hz), localizados antes e depois, respetivamente, de um segundo bloco de amplificação (novamente com o amplificador INA128). Desta forma, o sinal é amplificado e fica

<sup>10</sup> Os sinais de modo comum são desnecessários, visto que não transmitem qualquer tipo de informação, no fundo comportam-se como ruído, pelo que é importante reduzir ao máximo o seu ganho.

limitado a uma banda de frequências, eliminando alguns dos possíveis ruídos já mencionados em secções anteriores. A onda resultante surge representada, na Figura 2.12.

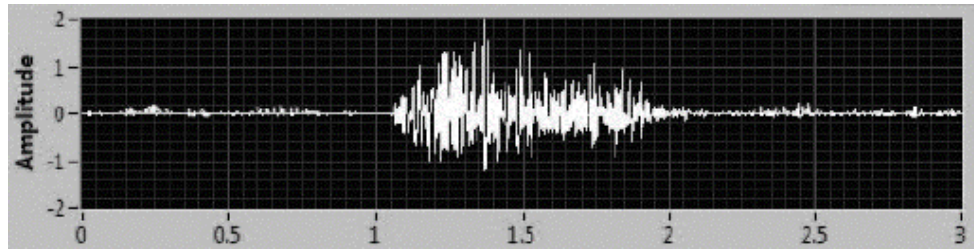


Figura 2.12: Exemplo 1 do sinal amplificado (Adaptado de [10]).

### 2.3.2.2. Modelo 2

O modelo seguinte, referente a [11], apresenta um circuito mais complexo, face ao anterior, como se vê pela Figura 2.13. O primeiro bloco em destaque, trata-se de um pré-amplificador, que recorre ao amplificador de instrumentação INA128, o qual recebe, nas suas entradas + e –, os sinais provenientes de dois eléktodos e, mais uma vez, para reduzir o ganho de modo comum, recorre-se a uma montagem com dois amplificadores operacionais OPA2604. Neste caso, realça-se ainda que, sendo a amplitude do sinal EMG baixa, por consequência a diferença em amplitude, entre a parte útil do sinal e o ruído, é bastante reduzida. Como tal, se o ganho deste pré-amplificador for demasiado elevado, corre-se o risco de, para além de amplificar o sinal eletromiograma, amplificar o ruído, levando a uma situação de instabilidade e saturação do próprio amplificador.

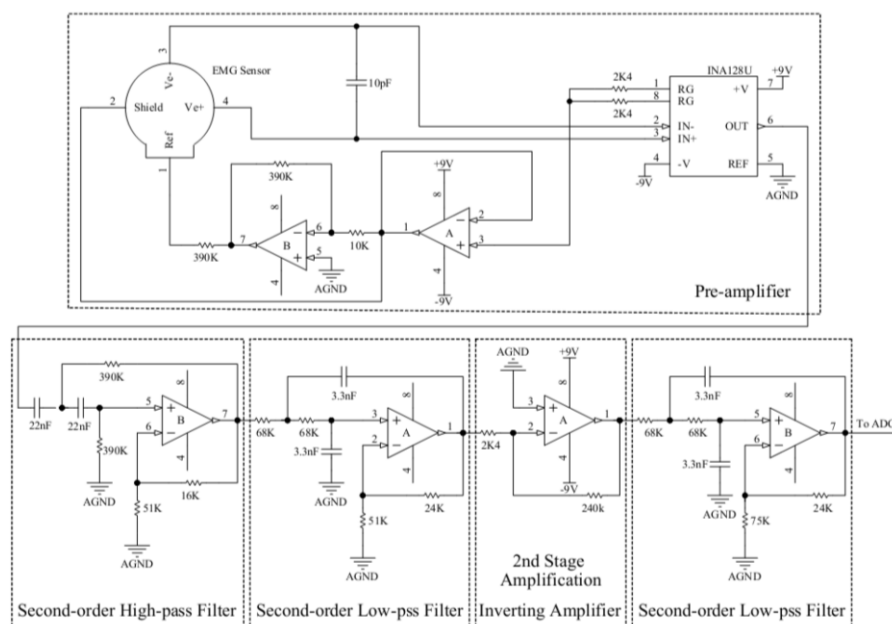


Figura 2.13: Exemplo 2 de modelo de amplificação biomédica (Adaptado de [11]).

De seguida, o sinal à saída do amplificador de instrumentação passa por dois filtros: o primeiro é um filtro passa-alto *Sallen-Key*<sup>11</sup> de segunda ordem ativo<sup>12</sup>, que utiliza os mesmos componentes do filtro passivo, mas que, ao acrescentar um amplificador, permite, para além de filtrar, neste caso frequências acima da frequência de corte, aplicar um ganho ao sinal de que resulta a amplificação da onda; o segundo filtro tem exatamente as mesmas características mas para uma função de passa-baixo, e garante também a amplificação do sinal.

A frequência de corte para estes dois blocos é calculada pela expressão (2.4).

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (2.4)$$

A onda é posteriormente amplificada, com recurso a um inversor, e filtrada novamente, com um filtro passa-baixo, resultando no sinal da Figura 2.14. Importa esclarecer, que se recorre duas vezes a filtros passa-baixo, porque o ruído proveniente da radiação eletromagnética se encontra principalmente nas altas frequências.

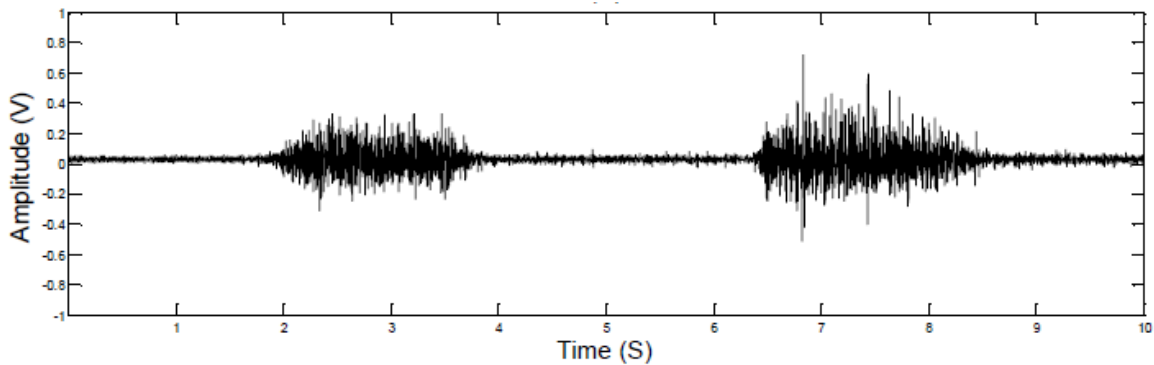


Figura 2.14: Exemplo 2 do sinal amplificado (Adaptado de [11]).

<sup>11</sup> *Sallen-Key* é uma topologia típica para filtros de segunda ordem.

<sup>12</sup> Um filtro ativo, caracteriza-se pelo uso de elementos ativos, como é o caso de amplificadores operacionais.

## 2.4. Processamento do Sinal Eletromiográfico

Passando a fase de condicionamento, importa definir os equipamentos necessários para efetuar o processamento do sinal, ou seja, é preciso analisar quais os métodos mais comuns para aquisição de dados, referentes ao eletromiograma, de modo a serem posteriormente trabalhados.

### 2.4.1. Conversor Analógico-Digital

Existem vários dispositivos com capacidade para desempenhar esta função, sendo fundamental a existência de um conversor analógico-digital, ADC (Figura 2.15), já mencionado no primeiro capítulo. Este tipo de conversor é de extrema relevância para este projeto, visto que, sendo o sinal muscular analógico, é necessário conseguir digitalizá-lo, para depois ser analisado com recurso a um *software*, próprio para o efeito.

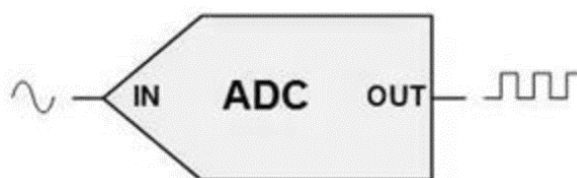


Figura 2.15: ADC (Adaptado de [12]).

Resumidamente, com a conversão, todos os valores analógicos são convertidos em valores binários, 0 e 1, sendo um processo que contempla duas fases.

A primeira fase é a denominada *sample and holding* e diz respeito à variação contínua no tempo do sinal, pelo que é necessário conseguir estabilizá-lo durante um curto período de tempo, para o mesmo poder ser amostrado. Isto, resolve-se com recurso a um circuito de *sampling and holding* que, como o nome indica, durante o período de tempo necessário para a digitalização, mantém o valor analógico estabilizado.

A segunda parte do processo, passa pelo *quantizing and encoding*. Ou seja, o ADC terá um circuito para quantizar e codificar o sinal já estabilizado. À saída do S/H, tem-se um determinado valor de tensão, ao qual é atribuído um valor numérico. De seguida, procura-se o valor mais próximo, dentro da gama de valores do quantizador em questão, existindo depois uma codificação desse número em binário para  $n$  bits, consoante a resolução do ADC.

## 2.4.2. Dispositivos de Aquisição de Dados

### 2.4.2.1. Placa de Som

A referência [5], apresenta como proposta para placa de aquisição dados, a placa de som de um computador (Figura 2.16). Cada computador tem placas de som com especificações distintas, pelo que a escolha deve assegurar a melhor integridade do sinal.

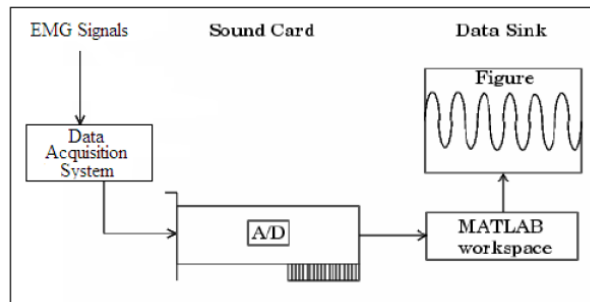


Figura 2.16: Placa de som (Adaptado de [5]).

Tipicamente, cada placa tem dois canais de *hardware*, que servem de input para os sinais mioelétricos, já amplificados e filtrados. Pela Figura 2.16, o eletromiograma é digitalizado com recurso a um ADC, contido na placa mas aqui importa realçar o recurso ao *software* MATLAB, para análise do sinal após digitalização. O programa *Simulink*, incluído no MATLAB, é o responsável por inicializar a placa de som para esta começar a receber os dados, como se constata na Figura 2.17.

Segundo [5], 95% da energia de um sinal muscular encontra-se entre os 10 e os 300 Hz, pelo que os filtros utilizados na parte do condicionamento do sinal, não limitaram a frequência nesta banda. Com recurso a este software, é possível utilizar filtros digitais, neste caso primeiro um filtro passa-alto, para permitir a passagem de frequências acima dos 10 Hz, em série com um filtro passa-baixo para permitir a passagem de frequências abaixo dos 300 Hz. Logicamente que o conjunto dos dois filtros, permite limitar a banda de frequências, acabando por atuar como um filtro passa-banda<sup>13</sup>. Note-se, ainda, que a maioria das placas de som funcionam numa banda entre os 20 e 220 kHz, o que implica que frequências abaixo dos 20 Hz sejam eliminadas, existindo, desta forma, atenuação dos sinais.

<sup>13</sup> Um filtro passa-banda, permite definir qual a banda de frequências que se pretende filtrar, eliminando as restantes.

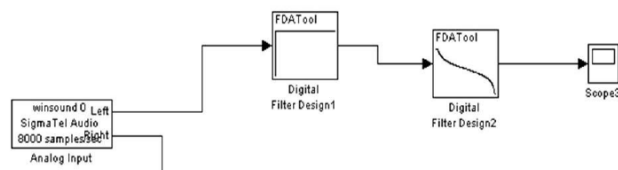


Figura 2.17: *Simulink* para placa de som (Adaptado de [5]).

#### 2.4.2.2. *Data Acquisiton System (DAQ)*

Recuperando [10] os sinais amplificados e filtrados no circuito apresentado anteriormente, esses sinais são posteriormente enviados para um computador, a funcionar em modo de bateria, através de um *Data Acquisiton System (DAQ)*, concretamente o NI USB-6009. Este dispositivo, trabalha com um *software* LabVIEW que, neste caso, com recurso a um filtro digital passa-banda, permite limitar a banda de frequências entre os 20 e 500 Hz (Figura 2.18).

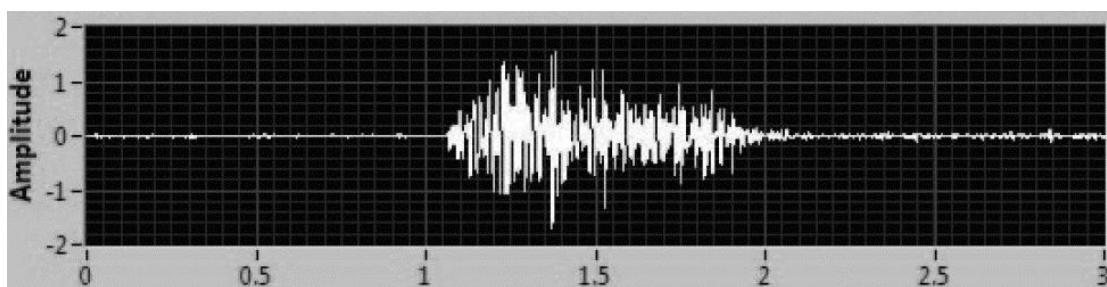


Figura 2.18: Sinal filtrado (Adaptado de [10]).

Conforme se retira de [10], verifica-se a necessidade de efetuar uma análise estatística do comportamento dos sinais mioelétricos detetados. Nos próximos capítulos, esta componente estatística estará em destaque e é de grande relevância para este tipo de projetos. Portanto, o passo seguinte passa pela retificação do sinal, ou seja, converter as partes negativas em positivas, com recurso à ferramenta de *squaring*, como se constata pela Figura 2.19.

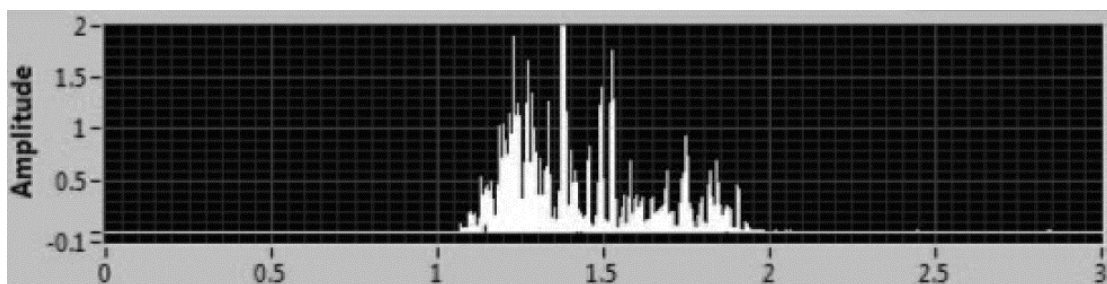


Figura 2.19: Sinal retificado (Adaptado de [10]).

Tendo isto em conta, os dados são depois enviados para um programa elaborado em MATLAB, para extrair oito características distintas, que se encontram representadas na Tabela 2.1.

Verifica-se, analisando a tabela, que foram usados quatro tipos diferentes de métodos estatísticos, como modos de análise, para os oito parâmetros em questão. Os dados registados serão, de seguida, classificados, conforme descrito no próximo capítulo deste projeto.

Tabela 2.1: Estatísticas do sinal mioelétrico (Adaptado de [10]).

Statistical Methods	EMG Features	Mean±SD							p-value	Predictor Importance
		HC	HO	WE	WF	FP	FS	AF		
ANOVA	IEMG	579.566 ±61.255	819.502 ± 43.750	594.419 ± 31.720	1421.52 ± 60.300	399.984 ± 67.928	553.062± 00.100	646.041 ± 64.313	0.000	----
	MAV	0.038 ± 0.004	0.054± 0.009	0.039± 0.008	0.094± 0.024	0.026± 0.004	0.036± 0.006	0.043± 0.004	0.000	----
	VAR	0.009 ± 0.001	0.021 ± 0.007	0.006 ± 0.003	0.046± 0.023	0.004± 0.001	0.004± 0.002	0.008± 0.002	0.000	----
	RMS	0.094± 0.009	0.145± 0.025	0.077± 0.020	0.209± 0.054	0.063± 0.010	0.066± 0.014	0.089± 0.011	0.000	----
	MNF	94.457 ± 4.345	81.351 ± 3.562	93.489± 4.297	67.732 ± 4.936	75.774 ± 2.273	82.365± 5.136	60.638± 2.987	0.000	----
CART	MNF	94.457 ± 4.345	81.351 ± 3.562	93.489± 4.297	67.732 ± 4.936	75.774 ± 2.273	82.365± 5.136	60.638± 2.987		1.0000
BT	IMEG	579.566 ±61.255	819.502 ± 43.750	594.419 ± 31.720	1421.52 ± 60.300	399.984 ± 67.928	553.062± 00.100	646.041 ± 64.313		1.0000
	MAV	0.038 ± 0.004	0.054± 0.009	0.039 ± 0.008	0.094± 0.024	0.026± 0.004	0.036 ± 0.006	0.043 ± 0.004		1.0000
	VAR	0.009 ± 0.001	0.021 ± 0.007	0.006 ± 0.003	0.046 ± 0.023	0.004 ± 0.001	0.004 ± 0.002	0.008± 0.002		0.9200
	RMS	0.094± 0.009	0.145± 0.025	0.077 ± 0.020	0.209± 0.054	0.063± 0.010	0.066 ± 0.014	0.089 ± 0.011		0.9200
	MNP	135.180± 25.804	328.100± 111.470	95.884 ± 49.717	701.510± 347.930	61.293± 19.649	69.259 ± 31.745	121.235± 30.950		0.9200
RF	MNF	94.457 ± 4.345	81.351 ± 3.562	93.489± 4.297	67.732 ± 4.936	75.770± 2.273	82.365 ± 5.136	60.638 ± 2.987		1.0000

## 2.5. Classificação do Sinal Eletromiográfico

Esta secção procura, abordar a temática da classificação do sinal muscular, com base nas suas características, resultantes da contração do músculo, e tendo como objetivo o respetivo registo e análise em termos estatísticos, como foi referido anteriormente. Neste caso, interessa perceber quais as ações que serão desencadeadas, quando são registados determinados padrões de um sinal. Trata-se de uma função relativamente complexa, pela diversidade e imprevisibilidade, visto que cada pessoa tem características físicas e biológicas bastante distintas e, por exemplo, é possível o mesmo indivíduo exercer mais ou menos força com o mesmo músculo.

Tendo em conta isto e seguindo [13], o principal método de classificação recorre à utilização de redes neuronais. Uma rede neuronal é um método, que evoluiu com base em modelos matemáticos de neurónios e sistemas de neurónios, sendo que nos últimos anos se tornou um método de grande utilidade para categorização de diversos tipos de dados.



### 2.5.1. Rede Neuronal Artificial

A rede neuronal artificial (RNA) é um modelo computacional inspirado no sistema nervoso central, mais em concreto no cérebro. A referida rede contém os chamados neurónios artificiais que se interligam entre si. Cada ligação, permite a transmissão de informação entre neurónios, sendo que o neurónio recetor recebe essa informação, processa-a e transmite a mesma aos neurónios a que está ligado. Desta forma, é possível efetuar uma aprendizagem, tal como acontece com o cérebro humano, sendo que numa rede neuronal cada neurónio e cada ligação tem um peso que é ajustado à medida que o processo de aprendizagem progride.

Importa referir dois conceitos presentes na referência [13], relativos às redes neuronais artificiais:

- *single layer perceptron* (SLP), que corresponde ao nível mais básico de uma rede neuronal (Figura 2.20), podendo somente classificar casos separados com um *target* binário (1,0).

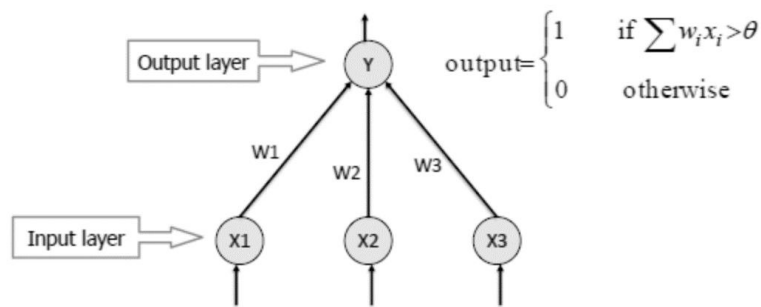


Figura 2.20: *Single layer perceptron* (Adaptado de [13]).

O SLP não tem conhecimento a priori, pelo que os pesos iniciais são determinados aleatoriamente. Assim, o algoritmo soma os produtos de cada input pelo seu peso e caso este valor seja superior a um determinado valor, o output fica a 1 (expressão (2.5)), caso seja inferior fica a 0 (expressão (2.6)).

$$w_1x_1 + w_2x_2 + \dots + w_nx_n > \theta \rightarrow 1 \quad (2.5)$$

$$w_1x_1 + w_2x_2 + \dots + w_nx_n \leq \theta \rightarrow 0 \quad (2.6)$$

Se o *output* previsto for o mesmo do desejado, então considera-se que o SLP tem um desempenho satisfatório e não é necessário recorrer a mudanças nos pesos. Porém, se isto não se verificar, então é necessário alterar os pesos para reduzir o erro, usando a expressão (2.7).

$$\Delta w = \eta \times d \times x \quad (2.7)$$

Sendo  $\eta$ , o fator de aprendizagem,  $d$ , output previsto – output esperado e  $x$  o valor de input.

- *multi-layer perceptron* (MLP), que tem por base a mesma estrutura do exemplo anterior, mas com uma ou mais camadas escondidas (Figura 2.21).

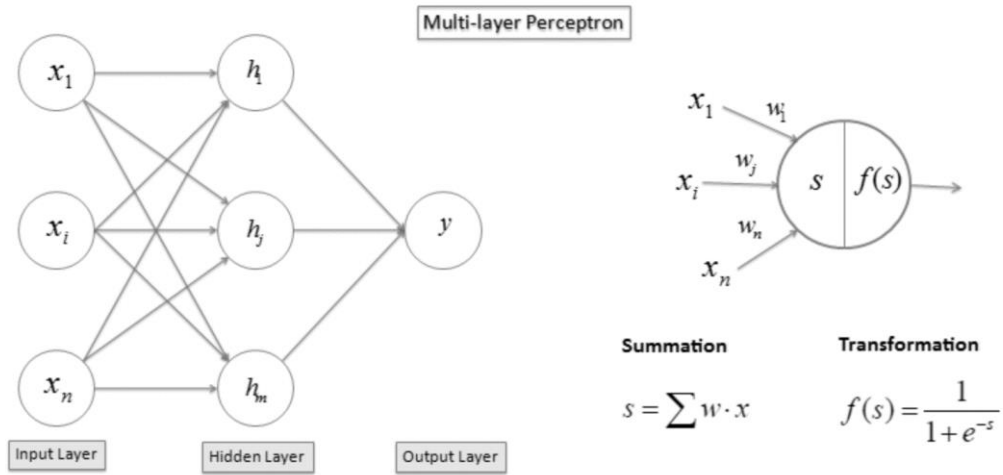


Figura 2.21: *Multi-layer perceptron* (Adaptado de [13]).

O algoritmo *backpropagation*, utilizado para o MLP, consiste em duas fases:

- *forward* – Propagação da soma do produto dos inputs pelo seu peso, seguido da computação dos outputs com recurso à função sigmoide, como se constata na Figura 2.21;
- *backward* – Propagação de erros, para trás seguindo a seguinte lógica:

1. Erro num neurónio de *output*:

$$d_0 = y \times (1 - y) \times (t - y) \quad (2.8)$$

2. Erro num neurónio de uma camada escondida:

$$d_i = y_i \times (1 - y_i) \times (w_i \times d_0) \quad (2.9)$$

3. Alteração do peso, com recurso à expressão (2.7).

A referência [14], indica que em 1993 foi proposto um sistema computacional de controlo em tempo real, baseado numa rede neuronal, para reconhecimento de padrões de EMG dos gestos dos utilizadores. O sistema consistia em dois modos de comunicação distintos, derivados do sinal muscular. O primeiro, é um sinal de controlo contínuo, proporcional ao esforço do músculo que permite controlar software (*sliders* e *scroll bar*), sendo o segundo modo de comunicação, o reconhecimento de gestos. Desta forma, é possível para o computador fazer escolhas discretas, tais como: seleções de menu ou a direção dos *sliders*, ao serem executados diferentes gestos.

De referir que, para este sistema computacional, o SLP foi treinado pelo algoritmo *Widrow-Hoff LMS*, enquanto o MLP foi treinado com recurso ao algoritmo de *backpropagation*. Este modelo provou ter uma taxa de sucesso bastante elevada, cerca de 95% de precisão em termos de classificação.

A Figura 2.22 ilustra, outra situação interessante de estudo, da referência [14].

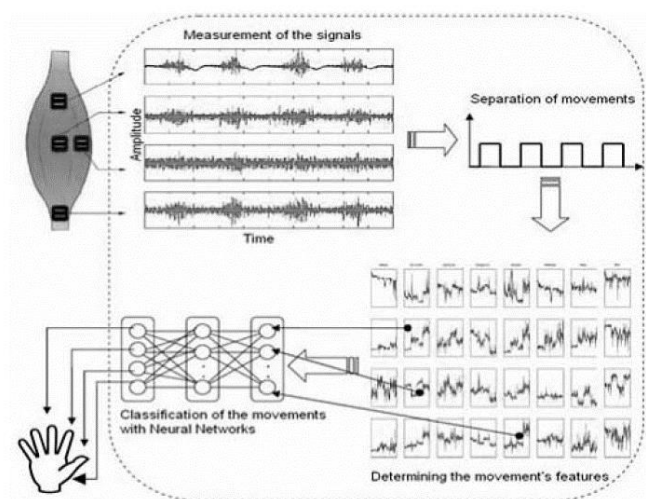


Figura 2.22: Classificação de sinais EMG para o movimento da mão (Adaptado de [14]).

Por volta de 2006, descobriu-se que a classificação dos movimentos da mão e dos dedos dependia significativamente da extração de características, o que é bastante relevante para melhorar a precisão da classificação. O procedimento de identificação foi descrito, com base nos padrões do eletromiograma da atividade do braço, utilizando vários modelos de redes neuronais.

Depois da comparação entre diferentes métodos computacionais inteligentes de identificação, chegou-se a uma taxa de precisão de 93%, usando MLP. Tal como se mencionou na secção anterior, estas características são de grande relevância, sendo que neste caso se destacam: *Mean Absolute Value* (MAV), *Variance* (VAR), *Waveform Length* (WL), *Norm*, *Number of Zero Crossings*, *Absolute Maximum*, *Absolute Minimum*, *Maximum minus Minimum* e *Median Value* (Med).

Importa referir que, logicamente, as características mais relevantes levam a uma maior e melhor precisão da classificação, embora muitas vezes haja dificuldade em perceber quais as que são mais ou menos importantes, para o caso em estudo.

## 2.6. Modelos de Aplicação

Embora o foco deste projeto se centre na ligação a um rato para computador e tendo em conta tudo o que já foi abordado, é possível encontrar diversos modelos distintos de aplicação para sinais mioelétricos, pelo que, de seguida, identificam-se alguns.

Diz a referência [15], que os sinais EMG podem ser usados para diversas aplicações, tais como: clínicas/biomédicas e interação máquina-homem. Em termos clínicos, pode servir como ferramenta para diagnósticos, permitindo detetar problemas neuromusculares, avaliar dores lombares ou distúrbios do ponto de vista motor.

A eletromiografia pode, também, ter como função detetar atividade muscular isométrica<sup>14</sup>, permitindo, desta forma, definir uma classe de gestos, subtis e sem movimento associado, destinados a controlar uma interface, sem perturbar o ambiente à volta e sem ser notado. Este dispositivo inclui um amplificador ligado aos elétrodos, com grande impedância na entrada, um filtro e um microcontrolador, para processar os sinais. Os resultados processados são transmitidos, com recurso a um módulo de comunicação *Bluetooth*. Concretamente, quando se verifica a ativação, o controlador envia um sinal wireless, para a principal unidade de processamento (por exemplo um telemóvel).

Um exemplo típico de outra situação em que ocorre interação homem-máquina, está presente num estudo feito pela NASA, relativamente à evolução dos seus sistemas de voo. O documento refere, essencialmente, a substituição dos *joysticks* de pilotagem, que dão os comandos ao computador de bordo, por sinais musculares obtidos por se ligarem vários elétrodos ao braço do piloto e este, através de movimentos musculares distintos, consegue reproduzir o comportamento real do *joystick*.

É interessante referir outro uso potencial para o eletromiograma, o qual permite verificar a existência de soluções com recurso a sinais mioelétricos, para casos de pessoas com paralisia ou com *deficit* significativo de mobilidade numa das mãos, como é comum acontecer na sequência de um AVC. Seja uma situação definitiva ou recuperável, é possível ultrapassar o referido *deficit*, como ilustrado na Figura 2.23. Note-se, que são os músculos do antebraço, que dão o comando à mão robótica que, por sua vez, ajuda a movimentar a mão do paciente.

---

<sup>14</sup> Tipo de atividade muscular que não se traduz em movimento.

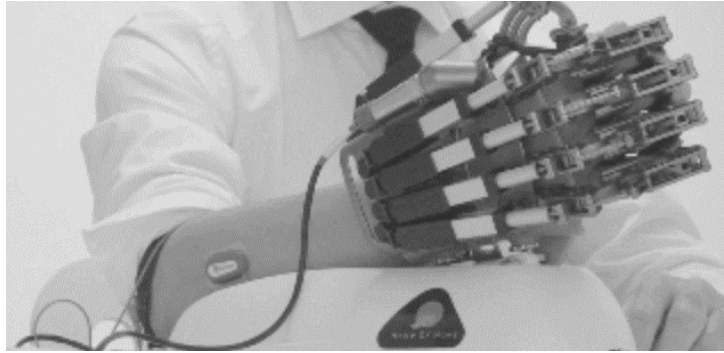


Figura 2.23: Mão robótica (Adaptado de [16]).

Noutra perspetiva, ainda dentro do tema saúde, o *Mime Speech Recognition* permite que indivíduos que perderam a capacidade de falar (p. ex., sem cordas vocais) consigam reproduzir a fala de forma indireta. Neste caso, o sistema reconhece um determinado discurso, por observação do músculo responsável pela fala. Ou seja, o resultado não é baseado em sinais áudio, mas sim em sinais mioelétricos. Tem uma grande vantagem: como não usa elementos sonoros, este método pode ser implementado em ambientes com muito barulho, sem afetar a correta interpretação do sinal.

# 3

## Testes e Análise de Resultados

A proposta de implementação deste protótipo, tem como base vários testes realizados desde o início da realização do projeto. Tendo em conta o descrito no capítulo anterior, facilmente se entende que todo o processo para a aquisição e tratamento dos dados do sinal mioelétrico precisa de vários tipos de testes diferentes, quer por necessidade de uma evolução na aprendizagem das características do sinal eletromiográfico, por parte do utilizador, quer por necessidade de encontrar quais os melhores métodos para conseguir atingir os vários passos, mencionados anteriormente, para movimentar o rato com os sinais musculares.

Assim, este capítulo abordará as várias etapas necessárias para descobrir a melhor solução de implementação deste protótipo, focando principalmente o material utilizado e os diversos testes feitos, tendo em conta, a amplificação e processamento do sinal.

### 3.1. Fase 1 – Captação do Sinal

A primeira fase de testes, consistiu, essencialmente, na tentativa de captação do sinal muscular, com recurso a músculos distintos, de forma a perceber as características de diferentes eletromiogramas, em termos de amplitude, neste caso tensão pico a pico<sup>15</sup>, e de forma de onda. Em concreto, foi necessário em primeiro lugar, escolher quais os elétrodos a utilizar, sendo que, pela análise feita no estado da arte, sabe-se de antemão que há uma limitação no tipo de elétrodo a utilizar, sendo obrigatório optar por elétrodos de superfície. Desta forma e após alguma pesquisa, entendeu-se que os modelos apresentados

---

<sup>15</sup> A tensão pico a pico exprime a diferença entre a amplitude máxima e mínima de uma forma de onda.

na Figura 3.1 e Figura 3.2 seriam opções viáveis, dado o aconselhamento por parte de especialistas em eletromiografias.

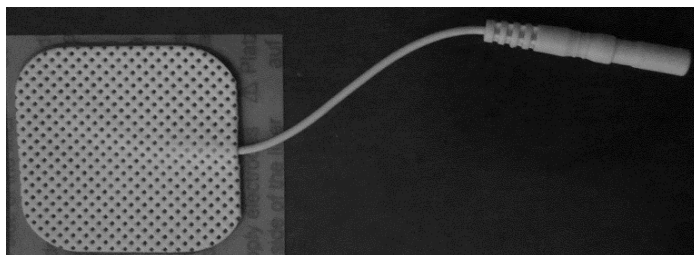


Figura 3.1: Eléctrodo de superfície Lifecare – 40X40MM.

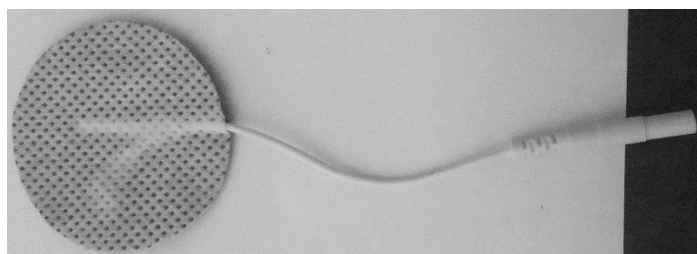


Figura 3.2: Eléctrodo de superfície Lifecare - 45MM.

Outra questão relevante, passa pela visualização da onda detetada pelos eléctrodos. Por questões de segurança, a ideia do osciloscópio “clássico”, típico de laboratório, foi abandonada, visto que há contacto direto, dos eléctrodos, com a pele e, logicamente, o osciloscópio é alimentado pela tensão eléctrica da rede (230V- 50Hz), havendo por isso risco de choque eléctrico. Por outro lado, quanto mais próximo de tomadas o utilizador estiver mais ruído proveniente da rede será detetado, o que não é desejável. Assim, o ideal é utilizar equipamento alimentado a baterias e em locais o mais isolado possível da rede eléctrica.

Desta forma, foi escolhido um dispositivo de aquisição de dados da National Instruments, NI myDAQ (Figura 3.3), que contém diversos instrumentos de laboratório *plug-and-play*, baseados no *software* LabVIEW, incluindo um multímetro digital, um osciloscópio, um gerador de funções, entre outros. É de notar, que este equipamento é ligado e alimentado por USB ao computador que, por sua vez, necessita de estar a funcionar em modo de bateria, visto que, caso esteja a carregar, influenciará o sinal detetado com ruído de alimentação.

Em consequência, para este primeiro teste, falta interligar os eléctrodos ao DAQ, sendo que, para tal, foram escolhidos, como condutores eléctricos para transmissão do sinal, cabos multifilares de cobre. Como esta placa de aquisição de dados tem duas entradas analógicas diferenciais (Figura 3.4) para o osciloscópio e escolhendo uma delas, neste caso AI0+ e AI0-, são, por isso, utilizados dois eléctrodos

para a captação da contração muscular. Para tal, um dos eléttodos deve estar colocado o mais ao centro possível do músculo pretendido, pelas razões mencionadas no capítulo anterior (este eléttodo ficará associado a AI0+), e o outro eléttodo deverá ficar posicionado numa zona do corpo afastado do primeiro, para que a respetiva tensão muscular seja praticamente nula (este eléttodo ficará associado a AI0-).

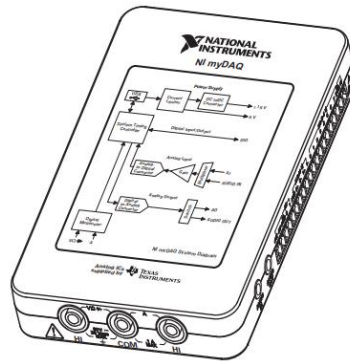


Figura 3.3: Dispositivo de aquisição de dados NI myDAQ - National Instruments (Adaptado de [17]).

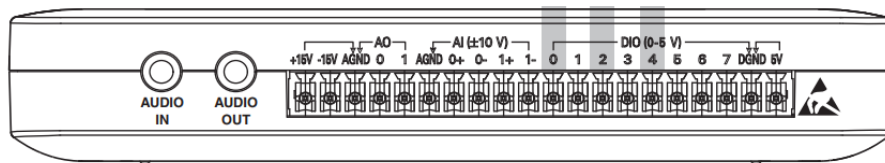
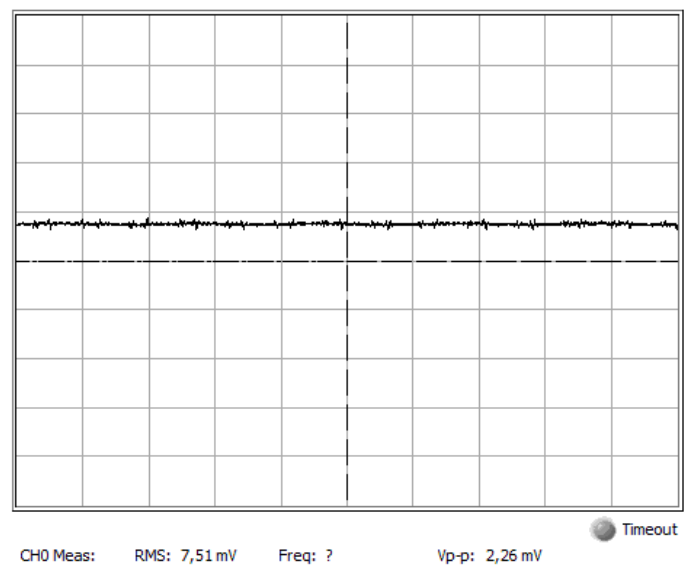


Figura 3.4: Inputs e outputs do dispositivo de aquisição de dados NI myDAQ (Adaptado de [17]).

Finalmente, foram analisados dois músculos distintos, para tentar perceber as características do sinal mioelétrico dos mesmos. Em primeiro lugar foi escolhido o bíceps, um músculo que fica entre o ombro e o cotovelo e, depois, o músculo frontal, que se encontra na testa. Em relação ao bíceps, na situação de repouso, em que não há contração muscular, verifica-se o sinal da Figura 3.5 no osciloscópio, sendo que, quando o músculo é contraído, o sinal detetado assume a forma presente na Figura 3.6.

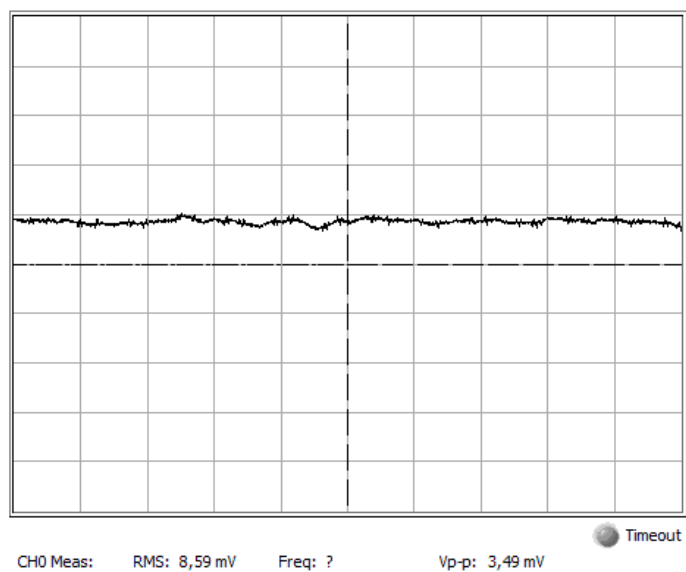
Comparando os dois gráficos, o movimento do músculo corresponde a uma ligeira alteração na configuração da onda. Verifica-se um aumento da tensão de pico a pico ( $V_{p-p}$ ), de 2,26mV para 3,49mV, que é bastante reduzido, pelo que não é possível ter a certeza absoluta se esta alteração do comportamento do sinal se deve à estimulação muscular, ou a outro fator externo, por exemplo, má fixação do eléttodo na pele.





Channel Settings	Channel 0 Values
Source	AI 0
Volts / Div (V)	10 mV
Vertical Position (Div)	0 (Div)
Parameter	Value
Device	myDAQ1 (NI myDAQ)
Time Base (s)	10 ms
Trigger Type	Immediate

Figura 3.5: Bíceps relaxado – Fase 1.



Channel Settings	Channel 0 Values
Source	AI 0
Volts / Div (V)	10 mV
Vertical Position (Div)	0 (Div)
Parameter	Value
Device	myDAQ1 (NI myDAQ)
Time Base (s)	10 ms
Trigger Type	Immediate

Figura 3.6: Bíceps contraído – Fase 1.

Relativamente ao músculo frontal, a Figura 3.7 apresenta a situação em que há ausência de movimento, sendo a Figura 3.8 a correspondente à contração. Tal como no caso anterior, ocorre um ligeiro aumento, de Vp-p, de 1,71mV para 4,71mV. Mais uma vez, a alteração na forma de onda pode dever-se a uma situação semelhante à anteriormente descrita, sendo, neste caso, ainda mais provável, pois a zona onde é colocado o eléctrodo é bastante propícia a uma má fixação do mesmo à pele, aquando da estimulação muscular.

Tendo em conta esta análise e como era esperado, com as amplitudes registadas não é possível estabelecer uma escala no osciloscópio que permita entender melhor os resultados obtidos, pelo que, tal como visto na secção 2.2., concluiu-se que seria essencial recorrer a um amplificador.

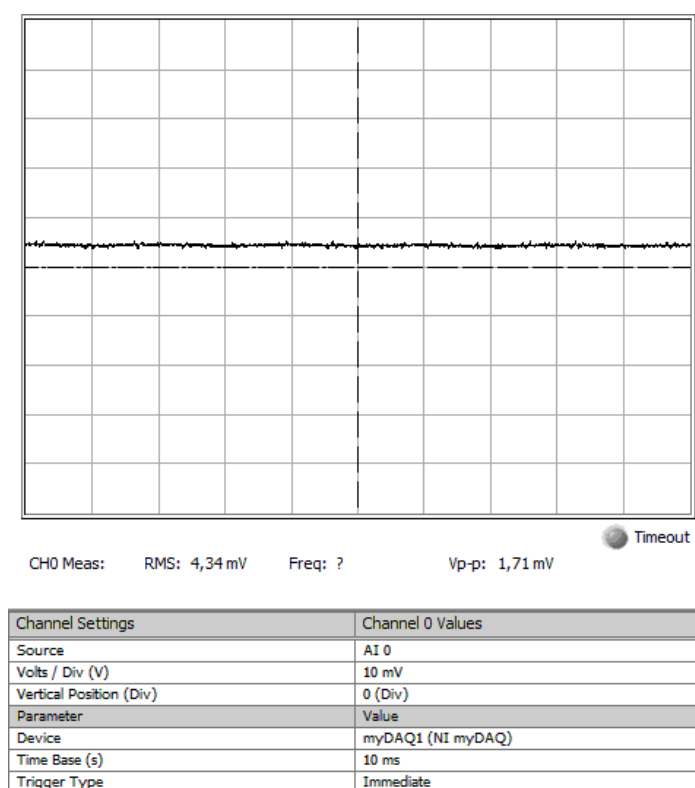


Figura 3.7: Frontal relaxado – Fase 1.

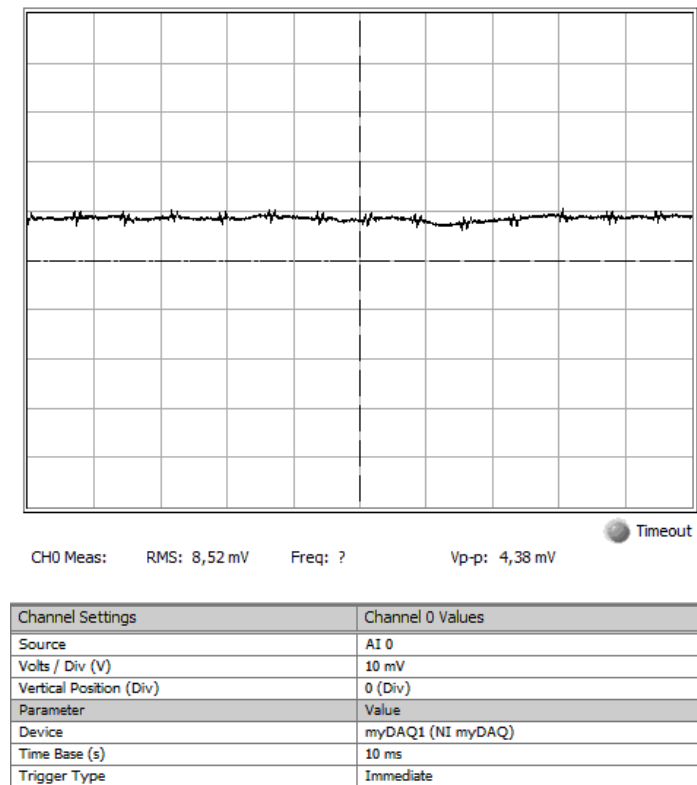


Figura 3.8: Frontal contraído– Fase 1.

### 3.2. Fase 2 – Circuito de Amplificação

Esta segunda fase pretende mostrar como foi feita a escolha do circuito de amplificação e apresentar os primeiros resultados obtidos, com a implementação desta montagem. Tendo em conta os modelos já vistos na revisão bibliográfica, deu-se relevância à constante utilização de amplificadores de instrumentação, pelo que a solução encontrada tem como base um amplificador deste tipo, que já foi detalhadamente abordado em 2.3.1., tendo sido escolhido o AD620.

A escolha deveu-se, sobretudo, a uma placa de amplificação (Figura 3.9), para sinais biomédicos, já previamente dimensionada para outro projeto académico, que foi reutilizada para este trabalho e cuja componente base é o amplificador de instrumentação acima mencionado.

Para perceber como funciona este sistema é importante desconstruí-lo, de forma a descrever qual a função de cada componente da placa em questão. Começando pelo AD620, a Figura 3.10, apresenta o diagrama de conexões do mesmo, sendo de destacar os oito *pins*, em função dos quais funciona todo este sistema de amplificação. Importa primeiro salientar dois *pins* +IN e -IN, ou seja, uma entrada diferencial e um *pin* OUTPUT, para a saída do sinal amplificado. Existem ainda dois *pins* Rg, que permitem definir o ganho do amplificador, dois *pins*, -Vs e +Vs, para a alimentação e um último REF, que funciona como referência.

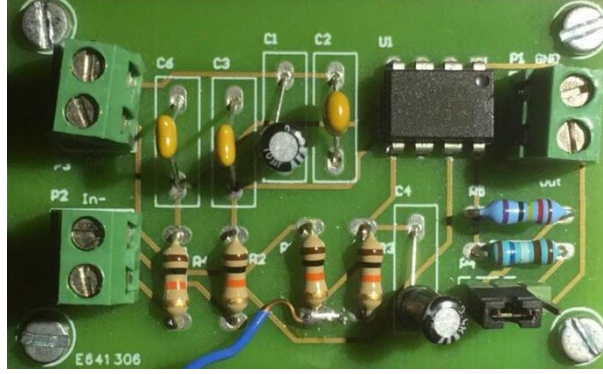


Figura 3.9: Placa de amplificação para sinais biomédicos.

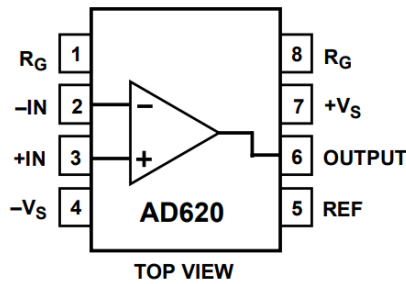


Figura 3.10: Diagrama de conexões do AD620 (Adaptado de [18]).

Entrando mais em detalhe, a Figura 3.11 ilustra o esquema simplificado do amplificador de instrumentação em causa, sendo que importa realçar que o *feedback* nos *loops* Q1-A1-R1 e Q2-A2-R2 mantém constante a corrente de coletor dos transístores Q1 e Q2, imprimindo a tensão de entrada na resistência externa, responsável pela seleção do ganho,  $R_G$ . Assim, é criado um ganho diferencial desde os *inputs* aos *outputs* de A1 e A2, dado pela expressão (3.1):

$$G = \frac{R1 + R2}{R_G} + 1 \quad (3.1)$$

Como R1 e R2, têm cada uma o valor de 24,7 k $\Omega$ , o ganho final é dado pela expressão (3.2):

$$G = \frac{49.4\text{k}\Omega}{R_G} + 1 \quad (3.2)$$

Neste caso, a placa de amplificação tem duas resistências (azuis) possíveis para atuarem como  $R_G$ , podendo o utilizador optar por uma ou por outra, consoante o ganho que deseje. Tem-se então uma

resistência de  $70,5\text{ k}\Omega$ , levando a um ganho de 1,7 e uma resistência de  $1,69\text{ k}\Omega$ , para um ganho de 30,23.

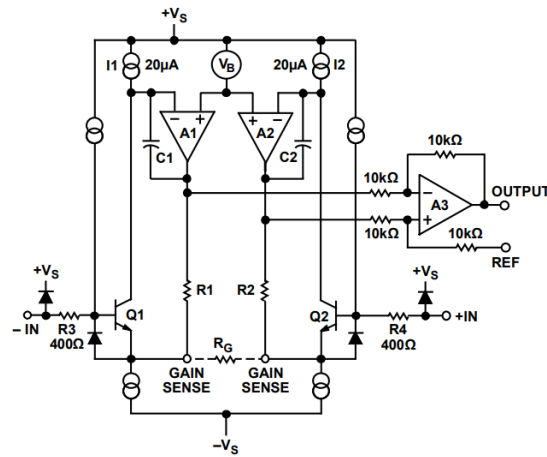


Figura 3.11: Esquema simplificado do AD620 (Adaptado de [18]).

Ainda relativamente à Figura 3.9, importa referir que a placa tem uma entrada diferencial, uma entrada para a tensão de alimentação, um pin para 5 V, ficando o outro pin ligado a *ground* e uma saída, um pin para o sinal, já amplificado, e outro que fica a *ground*. Pela Figura 3.12, verifica-se que, após a injeção dos sinais na entrada positiva e na entrada negativa, recorre-se a um filtro passa-baixo, para conseguir impedir a passagem de frequências acima dos 16 kHz e, conseqüentemente, eliminar algum ruído, presente nas altas frequências.

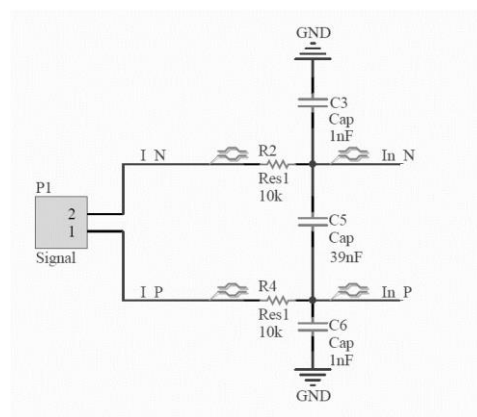


Figura 3.12: Filtragem dos sinais de entrada.

Para definir a tensão de referência, é ligado um elétrodo ao fio azul da Figura 3.9 e a análise do circuito da Figura 3.13 indica que  $V_{ref}$  terá o valor de 2,5 V, já que a alimentação do circuito é uma onda com amplitude máxima de 5 V e mínima de 0 V.

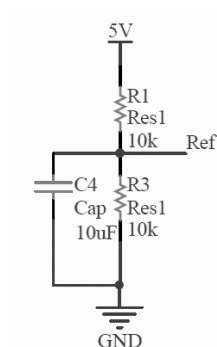


Figura 3.13: Circuito para a tensão de referência.

Por último, referir que este sistema de amplificação inclui condensadores de desacoplamento, colocados em paralelo com os *pins* de alimentação, sendo necessários em duas possíveis situações. Por um lado, no caso de uma queda na tensão de entrada, os condensadores fornecerão energia suficiente ao circuito, para manter a tensão estável. Por outro lado, havendo um aumento da tensão de entrada, os condensadores absorvem a energia em excesso, mantendo a tensão estável.

O teste a realizar implicou a utilização de três elétrodos: os dois elétrodos mencionados em 3.1 mais o elétrodo que serve de referência, atrás indicado, e que, neste caso, tem de ser colocado numa zona com poucas fibras musculares, por exemplo o cotovelo, em que a pele está praticamente em contacto com o osso. Portanto, os dois elétrodos da secção anterior serão ligados às entradas In+ (elétrodo do músculo que se pretende analisar) e In- (elétrodo colocado num músculo que não será contraído) do circuito de amplificação, ficando o elétrodo de referência conectado com a referência da montagem amplificadora. O *pin* de saída do sistema de amplificação (*Out*), referente ao sinal, ficará associada a AI0+, sendo o outro *pin* (Gnd), ligado a AI0-.

Para esta fase, as experiências efetuadas recorreram, exclusivamente, ao bíceps, sendo que os resultados foram todos bastante semelhantes. Desde logo se constata, na Figura 3.14, a situação do músculo relaxado e na Figura 3.15 o bíceps contraído<sup>16</sup>. Verificando os resultados, é de notar uma clara melhoria na captação do sinal já que, recordando os eletromiogramas do capítulo 2, a onda obtida no osciloscópio na Figura 3.15, aquando da estimulação do músculo, apresenta bastantes semelhanças com os gráficos da revisão bibliográfica. Obviamente que, neste caso, a escala não permite tanto ter essa perceção mas, reparando na tensão pico a pico, o amplificador permite a obtenção de valores em repouso, na ordem dos 15 mV, conseguindo atingir, na fase de contração, 566 mV.

<sup>16</sup> Nota: Os gráficos apresentados para esta fase foram obtidos com uma escala diferente em relação à secção anterior e posteriormente, ou seja, nas próximas secções verificar-se-á o mesmo. Em cada figura, está indicada a escala temporal (eixo do X) e a de tensão (eixo do Y).

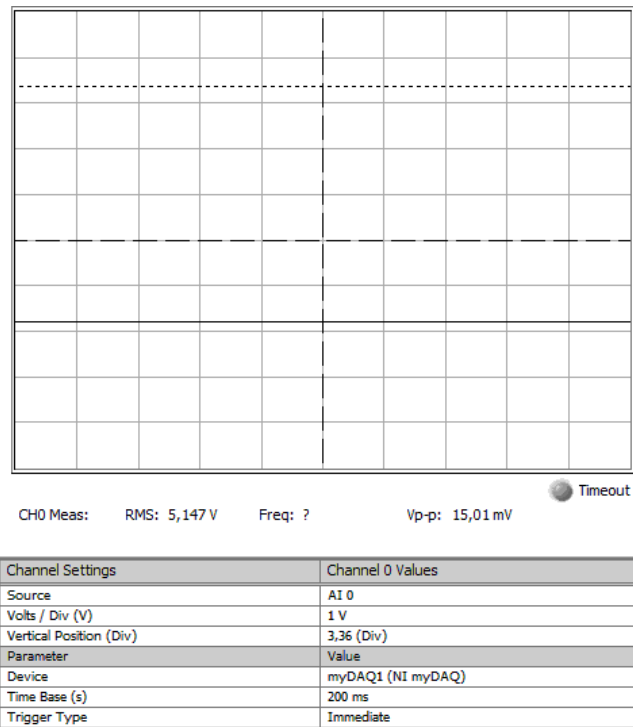


Figura 3.14: Bíceps relaxado – Fase 2.

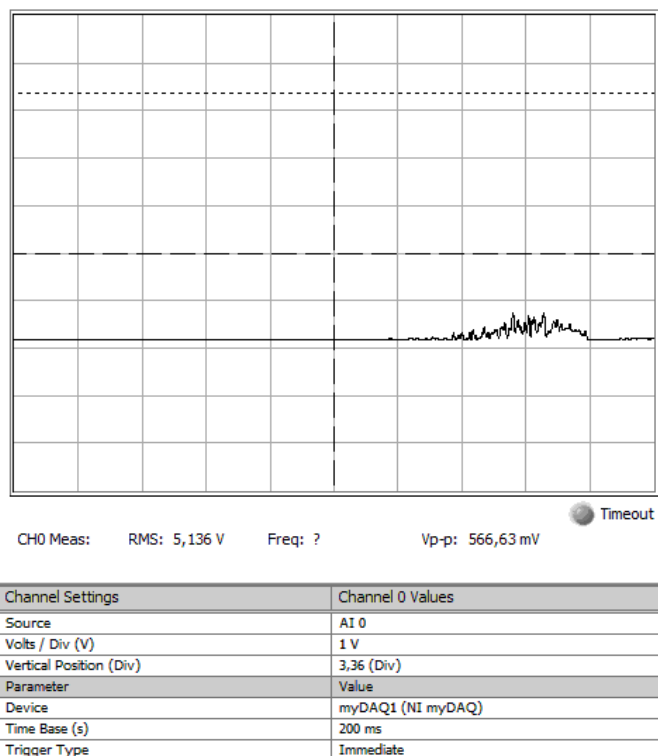


Figura 3.15: Bíceps contraído – Fase 2.

### 3.3. Fase 3 – Remoção de *Offset*

Ultrapassado o problema da amplificação do sinal, importa agora conseguir visualizar melhor o eletromiograma, principalmente para analisar, com mais detalhe, as características da forma de onda. No exemplo anterior, foi necessário recorrer a uma escala de 1 V/Div, por problemas de *offset*. Em concreto, o osciloscópio do NI myDAQ tem disponíveis escalas na ordem dos mV, que seriam mais interessantes para este tipo de análise mas, o facto de existir uma componente bastante acentuada de *offset* leva a que a onda não se fixe no ecrã. Sabendo então que o *offset*, corresponde à componente DC do sinal, foi necessário encontrar uma forma de remover a mesma.

Tendo em consideração que um sinal DC tem uma frequência de 0 Hz, uma forma bastante simples e eficiente para resolver este problema é recorrer a um filtro passa-alto de primeira ordem passivo, já abordado em 2.3, que permite a passagem de frequências acima dos 0 Hz, excluindo esta mesma. Daí que tenha sido escolhida, uma resistência de 5,1 K $\Omega$  e um condensador de 10  $\mu$ F, o que leva a uma frequência de corte, no valor de 3,12 Hz. Portanto, o filtro elimina a frequência indesejada, tal como se pretende.

Pensando, mais aprofundadamente, naquilo que se pretende com este projeto, tanto o bíceps como o músculo frontal, são bastante limitados no que diz respeito a uma diversidade de movimentos, ou seja, se o objetivo passa por implementar um rato para computador, são necessários vários comandos para conseguir movimentar o rato, em mais do que uma direção. Como tal, e no próximo capítulo serão abordados esses mesmos comandos, no que diz respeito aos movimentos necessários para os obter, optou-se por explorar um outro músculo responsável pela movimentação do pulso, denominado por músculo flexor radial do carpo (Figura 3.16).



Figura 3.16: Flexor radial do carpo do braço direito – Fase 3 (Adaptado de [19]).

Atentando na Figura 3.17, o flexor encontra-se numa situação de repouso, existindo posteriormente uma contração deste músculo (Figura 3.18), derivado do movimento do pulso. Analisando, note-se que no início a tensão de pico a pico se encontra nos 10 mV e posteriormente obtém-se uma onda com cerca



de 370 mV de Vp-p sendo que, com o efeito do filtro mencionado anteriormente, já é possível recorrer a uma escala mais adequada (50 mV). Diga-se, que a alteração na montagem, passa por ligar as saídas da placa de amplificação, ao filtro passa-alto e este, por sua vez, à entrada analógica do DAQ.

O sinal resultante que se observa confirma claramente aquilo que foi dito anteriormente ou seja, as características da onda obtida são muito semelhantes às descritas e visualizadas no Capítulo 2. Em concreto, percebe-se que o primeiro movimento do pulso leva a amplitudes mais elevadas, sendo que depois mantém-se a contração do mesmo, mas, com menos força, o que faz baixar consideravelmente as amplitudes registadas no osciloscópio.

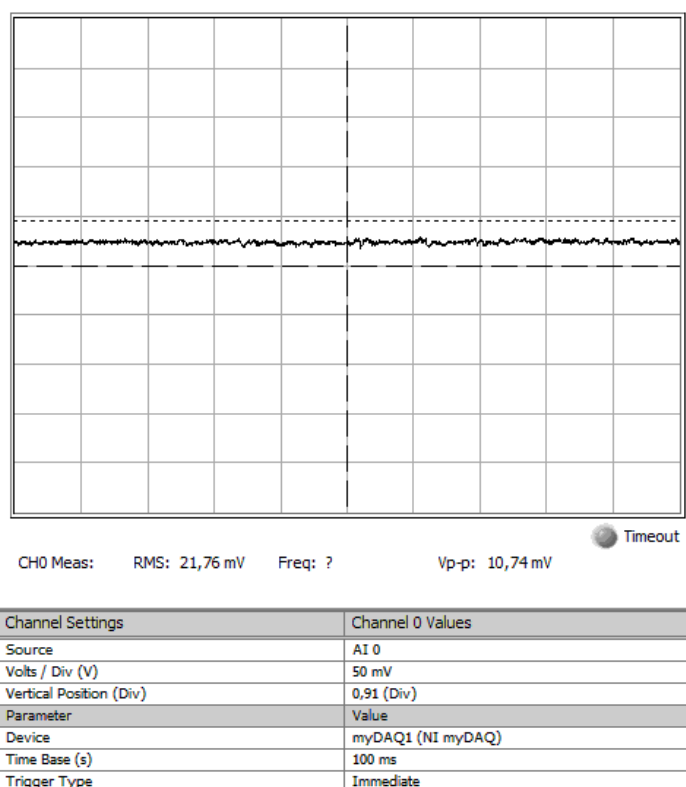


Figura 3.17: Flexor radial do carpo relaxado – Fase 3.

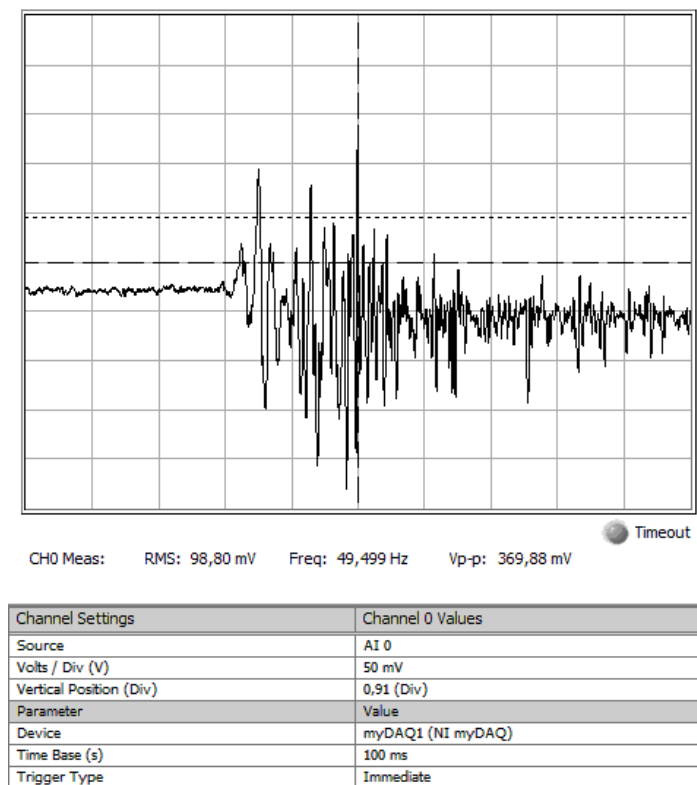


Figura 3.18: Flexor radial do carpo contraído – Fase 3.

### 3.4. Fase 4 – Níveis de Força

Estando definida esta estrutura, achou-se interessante, e mais à frente vai-se mostrar bastante relevante, perceber se era possível distinguir os sinais da contração do flexor radial do carpo, tendo em conta a força exercida, ou seja, estabelecer níveis de força e distinguir os sinais pelas diferenças de amplitudes registadas.

Os resultados permitiram verificar que tal é possível, pois a Figura 3.19 ilustra um eletromiograma com tensão Vp-p de 268 mV. Já na Figura 3.20 exerceu-se mais força aquando a contração do músculo e a tensão subiu para os 391 mV. Por último (Figura 3.21), um terceiro nível em que a força exercida foi ainda maior, face aos anteriores, e em que se obtiveram valores perto dos 560 mV.

Para além disto e não menos relevante, constatou-se, por estes três gráficos, que o sistema funciona de forma bastante eficiente, na medida em que, para além de existir uma resposta diferente, consoante o nível da força exercida no movimento do pulso, também há variação do sinal em resposta à respetiva duração.

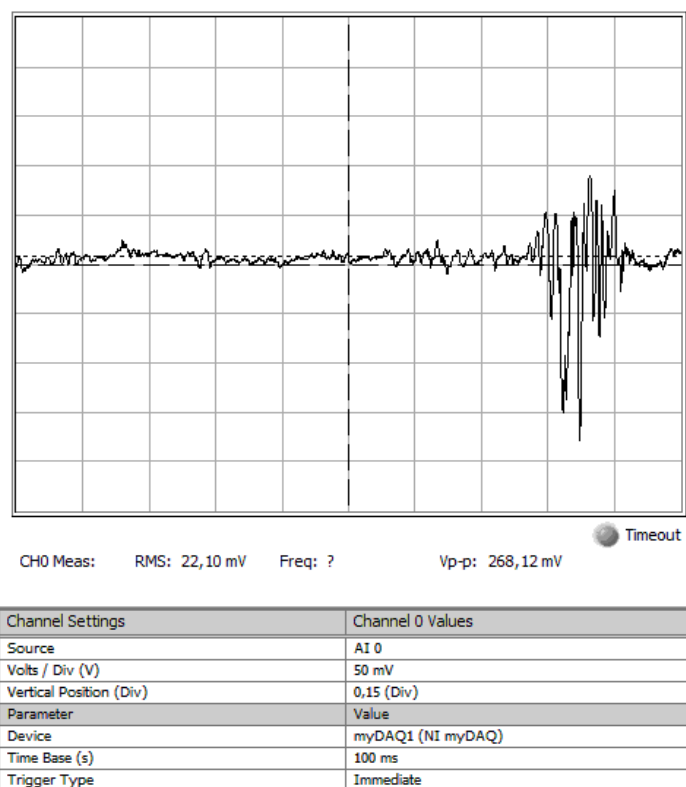


Figura 3.19: Contração nível 1.

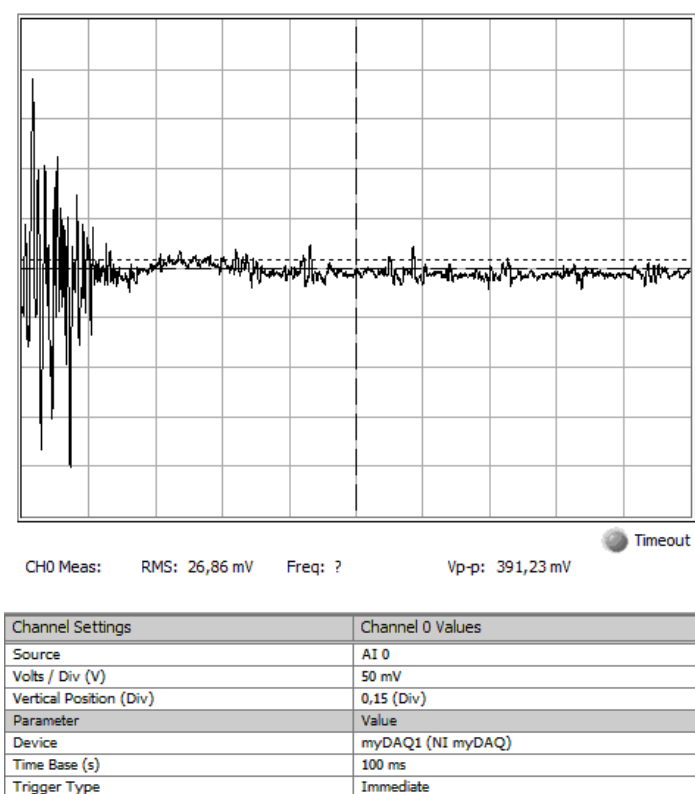


Figura 3.20: Contração nível 2.

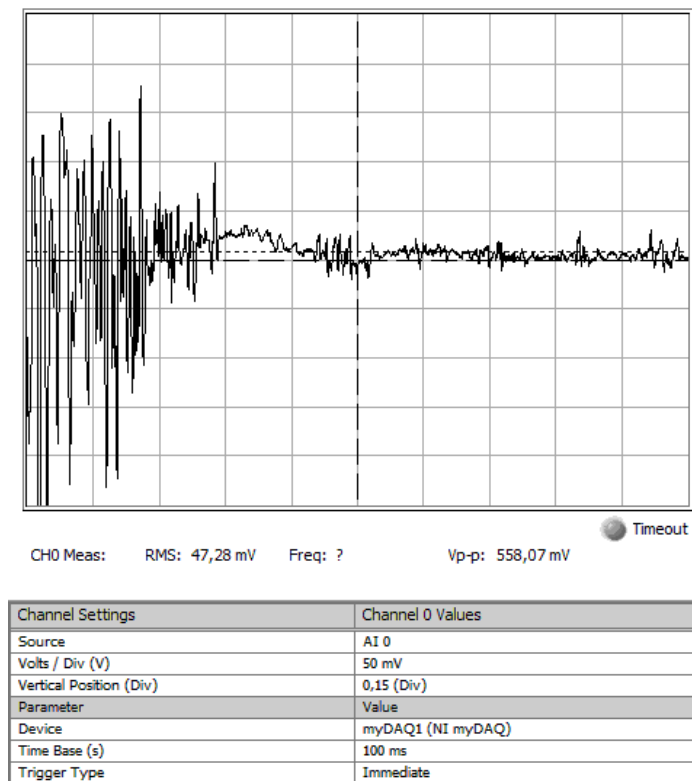


Figura 3.21: Contração nível 3.

### 3.5. Fase 5 – Processamento do Sinal com PSoC 5LP

A próxima etapa, seguindo a lógica do Capítulo 2, passa pelo processamento do eletromiograma. Para tal, foi decidido recorrer a um produto *low-cost* da Cypress, para prototipagem de produtos, que utiliza o processador ARM Cortex -M3 baseado no sistema PSoC 5LP, o CY8CKIT-059 PSoC 5LP Prototyping Kit (Figura 3.22). O PSoC 5LP, é o *programmable system-on-chip* mais eficiente da indústria, combinando periféricos digitais e analógicos programáveis, com o processador acima mencionado no mesmo chip.

Para esta escolha, foi tido em conta também o objetivo final deste projeto, ou seja, a implementação de um rato para computador. Portanto, pretende-se digitalizar o sinal, com recurso a um ADC, e programar este *kit* para funcionar como o pretendido periférico de entrada<sup>17</sup>.

Primeiramente, fez-se um estudo das possíveis implementações do ADC. Em concreto, procurou-se perceber quais os tipos de conversores analógico-digital disponíveis e as suas características, de forma a poder escolher a melhor opção para a implementação final. Existem dois tipos de ADC, na plataforma PSoC, o SAR ADC e o Delta\_Sigma ADC (Figura 3.23).

<sup>17</sup> Um periférico de entrada, como é o caso do rato, é um dispositivo que envia dados e sinais de controlo, para um sistema computacional.

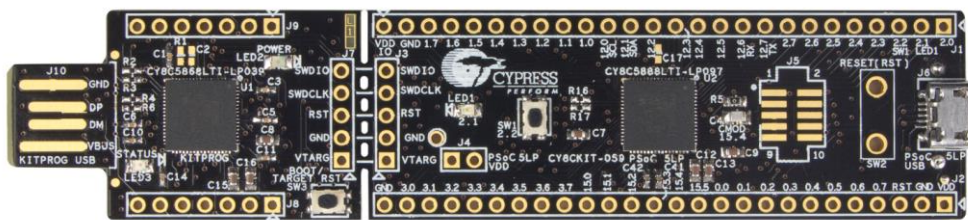


Figura 3.22: CY8CKIT-059 PSoC 5LP Prototyping Kit (Adaptado de [20]).

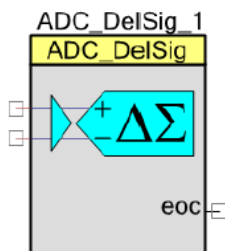


Figura 3.23: Delta\_Sigma ADC – PSoC 5LP.

Na secção 2.4.1, já foi abordado o comportamento típico deste sistema, mas importa referir que o dilema gerado em relação à escolha do modelo a utilizar, relaciona-se com a questão da taxa de conversão (também conhecida por frequência de amostragem) e da resolução do conversor. Em concreto, para determinados casos de estudo, é mais relevante ter uma maior resolução, em vez de uma frequência de amostragem relevante, e aqui há que referir que, quanto maior a primeira, menor será a segunda. Já noutros casos, pode-se verificar o oposto: é mais relevante ter uma taxa de conversão elevada do que a referida resolução.

Assim, por exemplo, se o projeto em causa estiver relacionado com comunicações, pretende-se sempre uma frequência de amostragem elevada, enquanto no âmbito deste trabalho, em que o importante é captar o sinal mioelétrico, com recurso a sensores (os elétrodos já referidos), é muito mais importante conseguir trabalhar com resoluções elevadas. Em consequência, optou-se pelo Delta\_Sigma ADC, visto que permite trabalhar com resoluções de 8 a 20 *bits*, estando o outro modelo referido, limitado a um intervalo de 8 a 12 *bits*.

No que concerne a digitalização do eletromiograma, os primeiros testes efetuados dizem respeito à Figura 3.24, sendo relevante explicar, para além do ADC, cada componente presente e a sua função no esquemático apresentado.

O sinal é injetado num dos vários portos do *kit*, sendo definido que esse porto é uma entrada analógica ADC\_In, que é ligado a um Delta\_Sigma ADC de 18 *bits*. Surge então um novo bloco, DMA (*Direct Memory Access*), que está dependente do conversor e importa detalhar. O DMA é utilizado para

transferir dados entre uma *source*, neste caso o ADC, e um determinado destino, sem intervenção do processador.

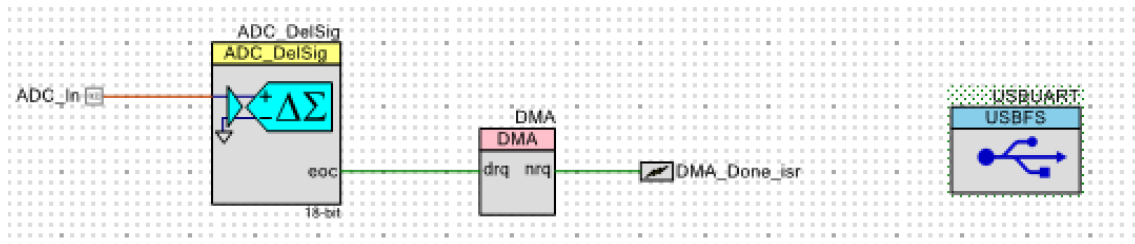


Figura 3.24: Esquemático para digitalização do sinal – PSoC 5LP.

Para tal, aquando do final de cada conversão, o conversor analógico-digital gera um sinal *eoc* (*end of conversion*), que serve de *trigger* para o terminal de entrada *drq* do DMA, permitindo a transferência de dados, ou seja, cada sinal *eoc* gerado permite a passagem de dados para um buffer, em termos de código, um vetor que é incrementado, à medida que recebe valores, havendo um limite de data que pode ser transferida por *eoc*, 2 bytes, e o buffer está limitado a 64 bytes. Desta forma, quando todas as posições do vetor estão preenchidas, é gerado um sinal pelo DMA, no seu terminal *nrq*, que ativa um *interrupt* *DMA\_Done\_isr*, o qual desativa o DMA.

A segunda parte deste processo, passa pela deslocação dos dados guardados no vetor para a componente USBFS, que funciona no modo USBUART e que importa perceber qual a sua função no sistema PSoC. É relevante referir que o objetivo nesta fase de pós-digitalização é conseguir visualizar o sinal, sendo que, para isso, o ideal é recorrer a uma ferramenta que permita a transmissão do sinal a uma velocidade considerável, de modo a não se sofrer perda de informação. Consequentemente, como o *kit* utilizado contém uma porta micro-USB, é possível criar protótipos com ligação USB 2.0 e, além disso, estabelecer uma comunicação UART (*Universal Asynchronous Receiver-Transmitter*).

Todo este processo é repetido constantemente e, para conseguir visualizar no computador o eletromiograma, recorre-se, novamente, a uma ferramenta da Cypress, o Bridge Control Panel. Basicamente, quando o PSoC 5LP é ligado por USB ao computador, a informação é transmitida para este programa por um COM (*Communication Port*), acabando por funcionar como um osciloscópio digital. O código que descreve este modelo exposto anteriormente, é apresentado em apêndice.

O gráfico da Figura 3.25 ilustra um teste inicial com esta montagem, em que se contraiu, de forma sucessiva o flexor radial do carpo, sendo interessante verificar que os sinais detetados, têm características bastante semelhantes aos sinais mioelétricos, apresentados anteriormente. Salienta-se que se verifica de forma significativa *offset*, já que o teste foi realizado sem recurso ao filtro passa-alto, que permite eliminar a componente DC, presente no EMG.

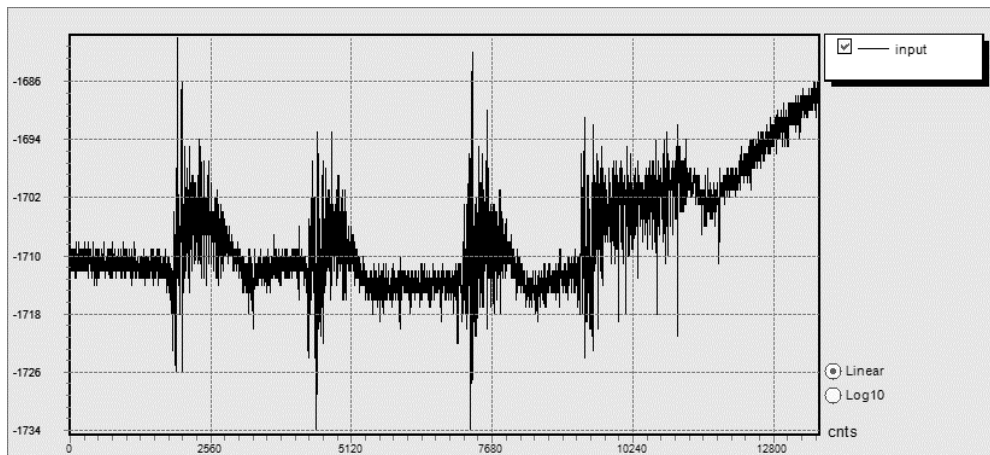


Figura 3.25: Contrações musculares digitalizadas – Bridge Control Panel.

Na Figura 3.26, consegue-se entender, ainda mais em detalhe, as semelhanças encontradas e verificar que, tal como nos testes anteriores, o sinal apresenta o comportamento esperado, já que se trata de uma contração mais duradoura, o que é bastante visível nesta imagem.

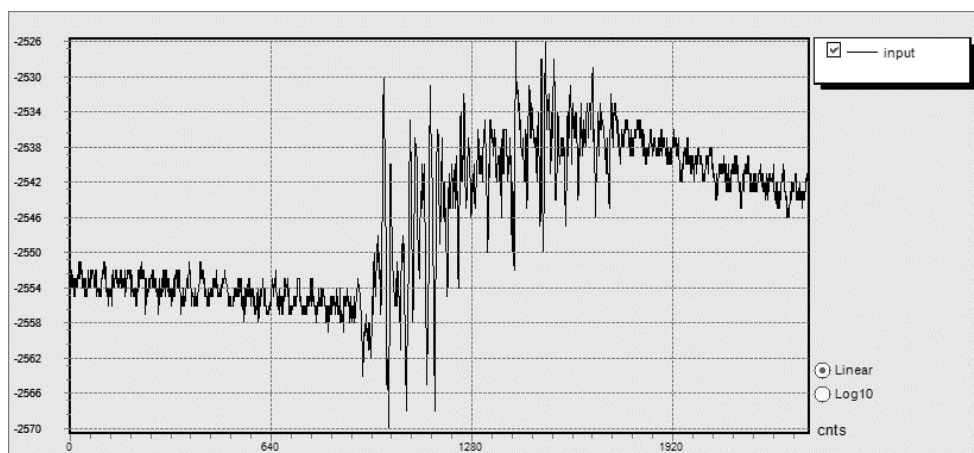


Figura 3.26: Contração muscular digitalizada– Bridge Control Panel.

### 3.6. Fase 6 – Processamento do Sinal com Matlab

Para finalizar, nesta etapa de testes foi aprofundado um pouco mais a parte de processamento do sinal já que, pela Figura 3.26, facilmente se verifica que o sinal, embora tenha a forma de onda pretendida, apresenta ainda bastante ruído e como o PSoC permite a implementação de filtros, procurou-se perceber qual a banda de frequências a eliminar, de forma a remover esse mesmo ruído.

Uma das vantagens do programa Bridge Control Panel é permitir a exportação dos valores do eletromiograma para um ficheiro, em formato de tabela. Tendo em conta isto, e antes de recorrer ao bloco filtro do PSoC, fez-se uma análise prévia de alguns dos eletromiogramas registados e importaram-se os dados desses sinais para a ferramenta Matlab.

A Figura 3.27, apresenta o EMG da Figura 3.26, enquanto *plot* do ficheiro Matlab, já com os dados importados, sendo que o eixo do X da Figura 3.26 é referente a contagens (número de pontos) e o da Figura 3.27 é relativo ao tempo, tomando em consideração, a frequência de amostragem do ADC, do PSoC. De referir que o código desta fase dos ensaios com recurso ao Matlab, encontra-se em apêndice, sendo este caso do sinal da Figura 3.27 apenas um dos vários testados.

Agora, com apenas algumas linhas de código em Matlab, é possível construir um filtro mas sem dispensar, primeiro, um estudo do espectro de frequências da onda da Figura 3.27. Ou seja, com recurso à função *fft* (*fast fourier transform*) do programa em questão, foi possível obter o gráfico da Figura 3.28. Note-se que o eixo do X apresenta os valores em Hz e o eixo do Y é relativo à transformada de Fourier do sinal captado, em escala logarítmica.

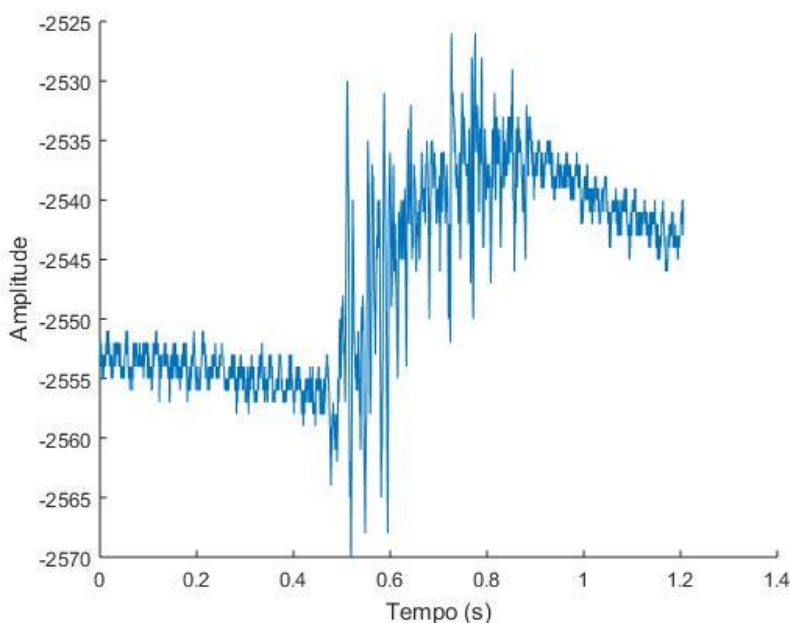


Figura 3.27: Contração muscular digitalizada – Matlab.

Para perceber melhor qual a banda de frequências do sinal muscular, é importante comparar o espectro da Figura 3.28 com o da Figura 3.29 visto que, a primeira diz respeito, primeiro a um momento sem contração muscular e depois com contração muscular e a segunda remete para o sinal captado sem movimentação do músculo.



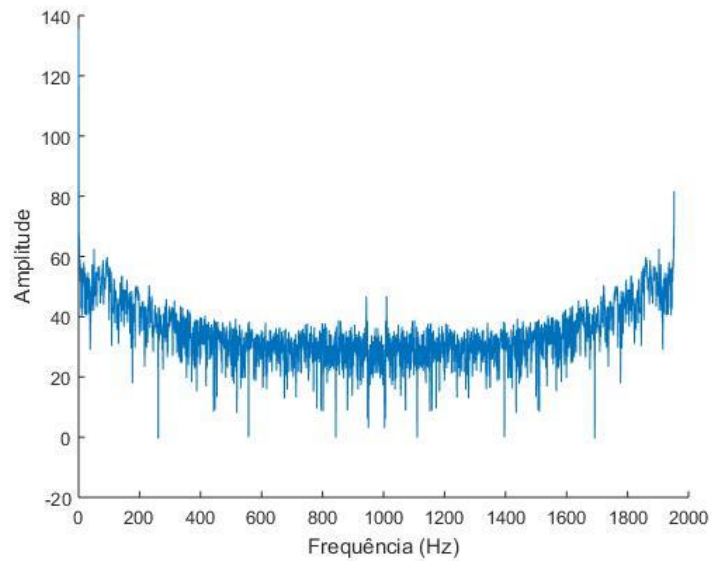


Figura 3.28: Espectro de frequências com contração muscular – Matlab.

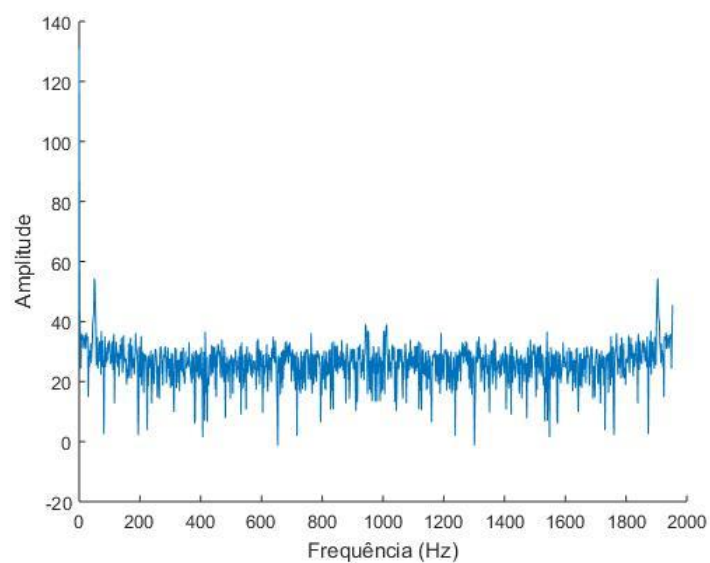


Figura 3.29: Espectro de frequências sem contração muscular – Matlab.

Comparando as duas figuras, destaca-se em ambas a presença de uma componente DC bastante acentuada, como já tinha sido referido. O relevante a retirar é que, no caso do espectro do eletromiograma, os valores da fft, no intervalo de frequências entre 10 e 100 Hz, são consideravelmente superiores aos do espectro sem contração muscular.

Foi implementado então o filtro, através do seguinte código:

```
Wn=[10 50]/(1953/2)

a = fir1(100,Wn, 'bandpass')

freqz(a,1)

figure (4)

y2 = filter(a,1,input1)

plot (time,y2)
```

A variável Wn diz respeito à banda de frequências, em que se pretende aplicar o filtro, sendo que, o Matlab trabalha com frequências normalizadas, logo é necessário dividir o intervalo [10 50] (dentro da gama mencionada anteriormente, entre 10 e 100 Hz) por metade da frequência de amostragem (1953Hz), ou seja, pela frequência Nyquist. A função fir1, que desenha um filtro do tipo FIR, apresenta como parâmetros a ordem do filtro (100), Wn, e o tipo de filtro, neste caso *bandpass* (passa-banda). O primeiro *plot* (Figura 3.30) destas linhas de código diz respeito à função freqz, que corresponde à resposta em frequência do filtro FIR.

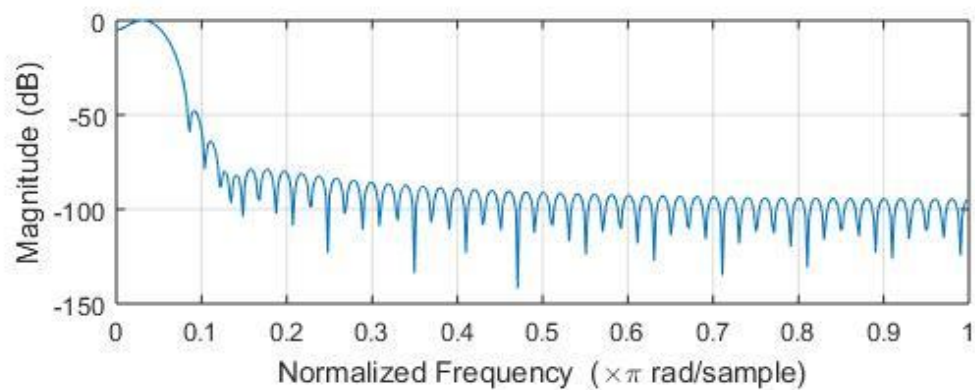


Figura 3.30: Resposta em frequência do filtro FIR – Matlab.

O efeito do filtro em questão pode ser verificado na Figura 3.31. Salienta-se que a forma de onda está bastante mais perceptível, tendo sido eliminado grande parte do ruído, pelo que o intervalo de frequências entre 10 e 50 Hz foi adotado como o filtro passa-banda da implementação final.

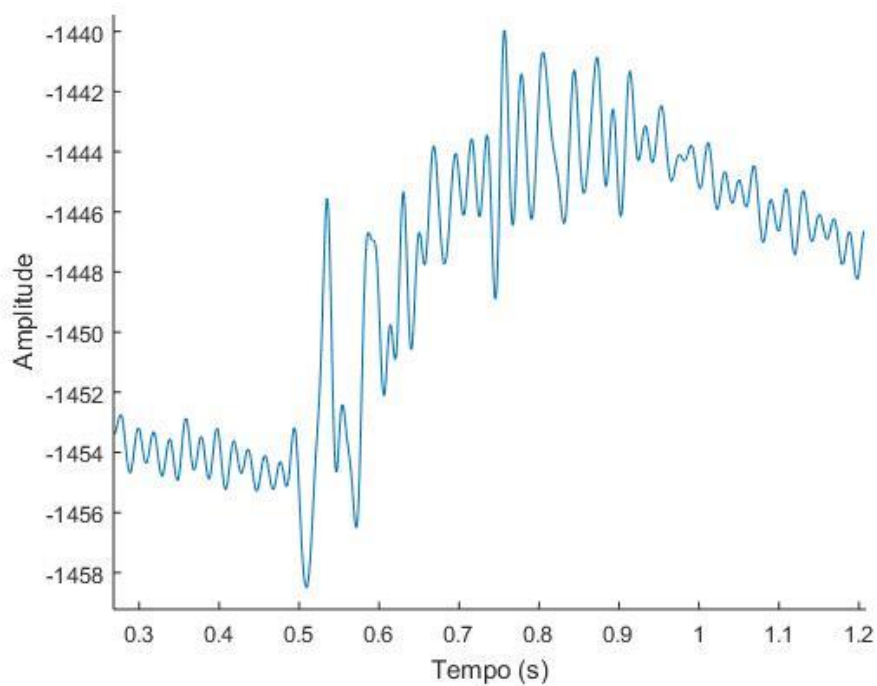


Figura 3.31: Contração muscular digitalizada com efeito do FIR– Matlab.

Pela Figura 3.32, sobrepondo a Figura 3.27 e a Figura 3.31, constata-se com maior nitidez as alterações produzidas pelo filtro.

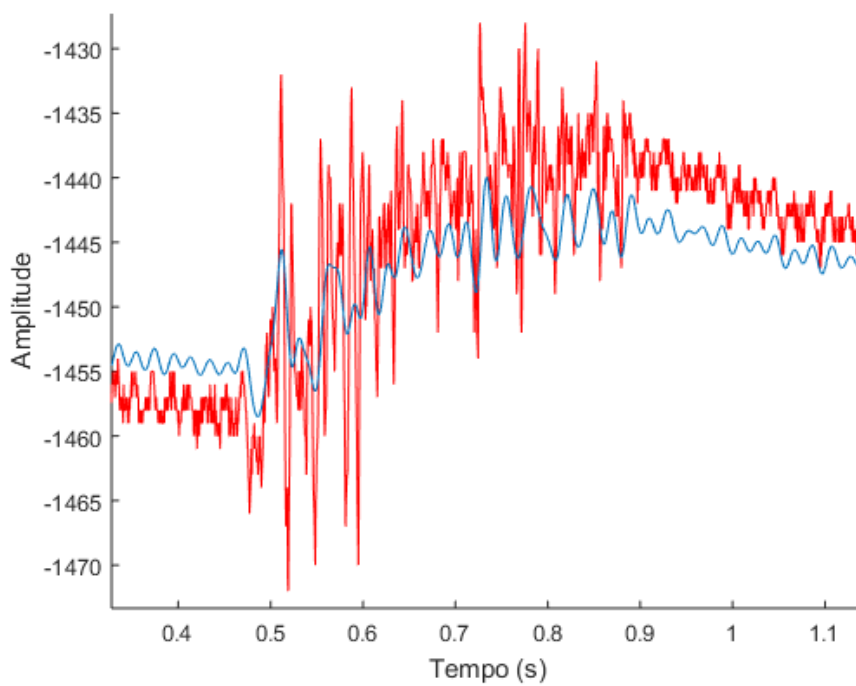


Figura 3.32: Contração muscular digitalizada com filtro (azul), sem filtro (vermelho).

# 4

## Implementação Final

Concluídos o estudo prévio no Capítulo 2, referente às metodologias existentes, e, no Capítulo 3, o desenvolvimento da aquisição, amplificação e processamento do sinal que controlará o rato do computador, é possível aplicar agora o conhecimento adquirido e implementar uma solução para o problema em questão. O que será abordado de seguida é uma proposta de implementação mas sem excluir que poderão existir outras alternativas.

De acordo com a secção 3.3, ficou definido que a melhor alternativa para exercer os comandos do rato seria recorrer ao músculo flexor radial do carpo, visto que é responsável pelo movimento do pulso. Por consequência, a diversidade de movimentos do pulso, permite definir vários comandos diferentes para aquele periférico de entrada, tendo sido definidos quatro movimentações distintas.

Assim, conforme indicado na Figura 4.1, tem-se:

- a) A contração do pulso para o lado direito, que coincidirá com a deslocação do rato para a direita;
- b) A contração para baixo, originando o movimento do rato para baixo;
- c) O flexor, ao contrair o pulso para cima, vai permitir movimentar o rato no ecrã para cima;
- d) A contração do pulso para a esquerda leva à deslocação do rato no mesmo sentido.

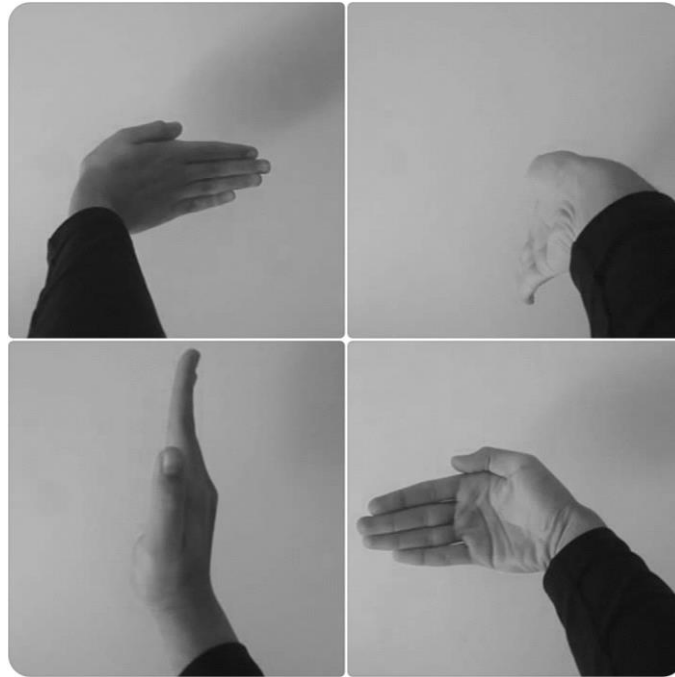


Figura 4.1: Comando: a) direita; b) baixo; c) cima; d) esquerda.

O Capítulo 3, em que se procurou descrever os vários testes realizados, termina com uma análise do sinal eletromiográfico, cujo processamento recorreu a Matlab, foi digitalizado com recurso ao sistema PSoC e visualizado com o Bridge Control Panel. Contudo, embora esta última opção seja bastante viável e interessante, para recorrer a este programa é necessário utilizar a componente de micro-USB do *kit* utilizado neste projeto. Como, mais tarde, a programação do rato será feita pela mesma porta USB, seriam necessários dois *kits*, para se poder visualizar, em simultâneo, o sinal mioelétrico e o movimento do rato. Logicamente, para conseguir deslocar o rato no ecrã, não é necessário, em simultâneo, observar o eletromiograma, mas acaba por servir de complemento quer para uma futura demonstração do funcionamento do protótipo, quer para efeitos de auxílio na programação e calibração do rato.

Neste sentido, optou-se por uma outra solução, abdicando da ferramenta Bridge Control Panel, e que permitiu conjugar as duas componentes mencionadas no parágrafo anterior, ao mesmo tempo, sendo, por isso, necessário analisar a Figura 4.2. Resumidamente, passa por digitalizar o sinal e, como se pretende o mínimo de ruído possível no EMG em causa, recorre-se a um filtro, com as características do filtro da secção 3.6, sendo o sinal filtrado enviado para um DAC (*Digital to Analog Converter*). Ou seja, os dados são novamente convertidos numa onda analógica, estando a saída associada a um dos *pins* do *kit*, com a vantagem da porta micro-USB ficar disponível para ser configurada, para atuar como rato, através da componente USBFS. Complementarmente, o eletromiograma é visualizado com recurso ao osciloscópio do NI myDAQ, referido no Capítulo 3.

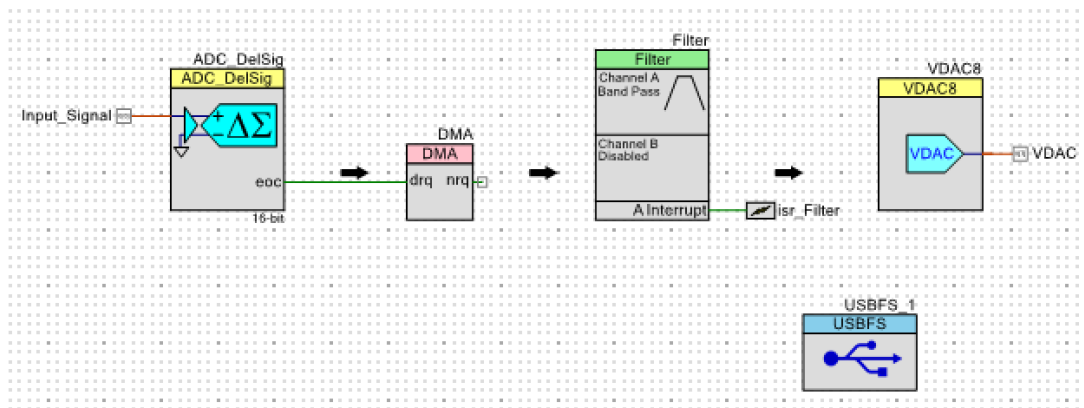


Figura 4.2: Esquemático da implementação final.

Importa agora explicar ao detalhe cada uma das componentes em questão nesta implementação e como estas se relacionam umas com as outras, tendo em conta o código necessário para o efeito. Na situação atual, o sinal é captado recorrendo aos dois elétrodos, já referidos anteriormente, havendo a posterior amplificação na placa de amplificação do Capítulo 3. A saída deste circuito fica ligada a um dos *pins* do *kit* PSoC, que se denominou por *Input\_Signal* ligado, por sua vez, à entrada do Delta-Sigma ADC. A Figura 4.3 apresenta as principais características e configurações estabelecidas para o ADC.

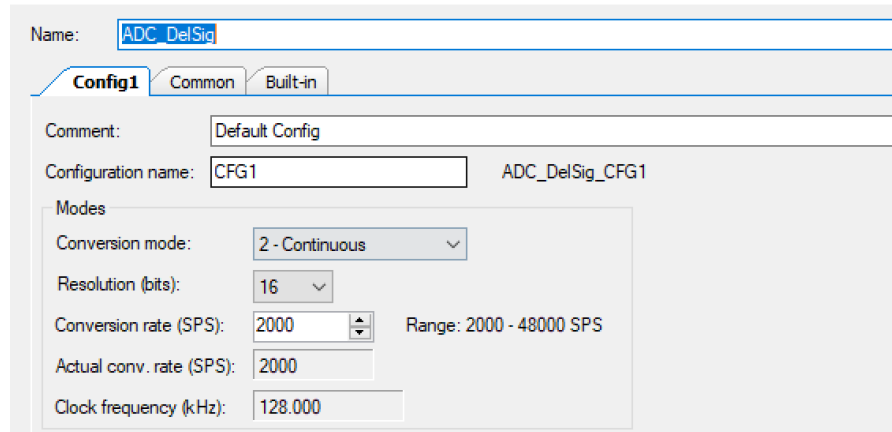


Figura 4.3: Parâmetros do Delta\_Sigma ADC.

Destaque-se alguns parâmetros de maior relevância, nomeadamente o *Conversion mode* que foi programado como *Continuous*. Ou seja, de entre os vários modos de conversão disponíveis, optou-se pelo modo contínuo, já que se trata do modo mais recorrente, quando se trabalha com um único sinal (caso deste projeto), estando o seu funcionamento dependente do *Conversion rate*, aqui definido com o valor de 2000 SPS, num intervalo possível entre 2000 e 48000 SPS. Em concreto, e como referido na secção 3.5, pretende-se uma resolução maior do ADC para uma frequência de amostragem menor. Portanto, tem-se um conversor com uma resolução de 16 *bits*, para uma taxa de conversão de 2000

amostras por segundo. Registe-se, ainda, que o parâmetro *Clock frequency* tem o valor de 128 kHz, sendo ajustado consoante os valores dos outros parâmetros já mencionados.

Uma das diferenças para o exemplo da secção 3.5, diz respeito à introdução de um filtro passa-banda. Note-se que, de novo, se recorre ao bloco DMA, sendo que neste caso os dados são transmitidos não para um vetor definido no código mas para um dos registos do filtro em questão, como se constata na seguinte função:

```
void DMA_Config(void)
{

#define DMA_BYTES_PER_BURST      (2u)
#define DMA_REQUEST_PER_BURST   (1u)
#define DMA_SRC_BASE             (CYDEV_PERIPH_BASE)
#define DMA_DST_BASE             (CYDEV_PERIPH_BASE)
#define TRANSFER_COUNT           (2u)

/* Variable declarations for DMA */
    uint8 channelHandle;
    uint8 tdChanA;

/* Configure the DMA to Transfer the data in 1 burst with individual
trigger for each burst.*/
    channelHandle = DMA_DmaInitialize(DMA_BYTES_PER_BURST,
DMA_REQUEST_PER_BURST, HI16(DMA_SRC_BASE), HI16(DMA_DST_BASE));

    /* This function allocates a TD for use with an initialized DMA channel
*/
    tdChanA = CyDmaTdAllocate();

    /* Source and Destination address increments are not needed as we are
using 2 byte transfers and Spoke Width is 16 bit */
    CyDmaTdSetConfiguration(tdChanA, TRANSFER_COUNT, tdChanA, 0u);

    /* Set the source address as ADC_DelSig and the destination as Filter
Channel A*/
    CyDmaTdSetAddress(tdChanA, LO16((uint32)ADC_DelSig_DEC_SAMP_16B_PTR), LO16((u
int32)Filter_STAGE_A_PTR));

    /* Set tdChanA to be the initial TD associated with channelHandle */
    CyDmaChSetInitialTd(channelHandle, tdChanA);

/* Enable the DMA channel represented by channelHandle and preserve the TD
*/
    CyDmaChEnable(channelHandle, 1u);
}
```

Esta função é gerada automaticamente pelo sistema PSoC, aquando a utilização de um DMA, correspondendo à configuração do mesmo, destacando-se a definição do endereço (CyDmaTdSetAddress) de origem, representado por um apontador para o ADC de 16 *bits* e o endereço de destino, através de um apontador para o registo A do filtro.

Agora, em vez de se ativar o *interrupt* no terminal nrq, o mesmo só é ativado após a atuação do filtro passa-banda, cujas características se constatarem na Figura 4.4. Tal como no filtro elaborado em Matlab (secção 3.6), recorreu-se à classe FIR e a uma banda de frequências entre os 10 e os 50 Hz. Ou seja, tratando-se de um filtro passa-banda, para um valor de frequência centrado em 0,03 kHz, é necessária uma largura de banda de 0,04 kHz. Verifica-se ainda que, tal como mencionado anteriormente, quando o número de *bytes* transferidos pelo DMA é processado pelo filtro ativa-se um *Interrupt request (Data ready signal)*.

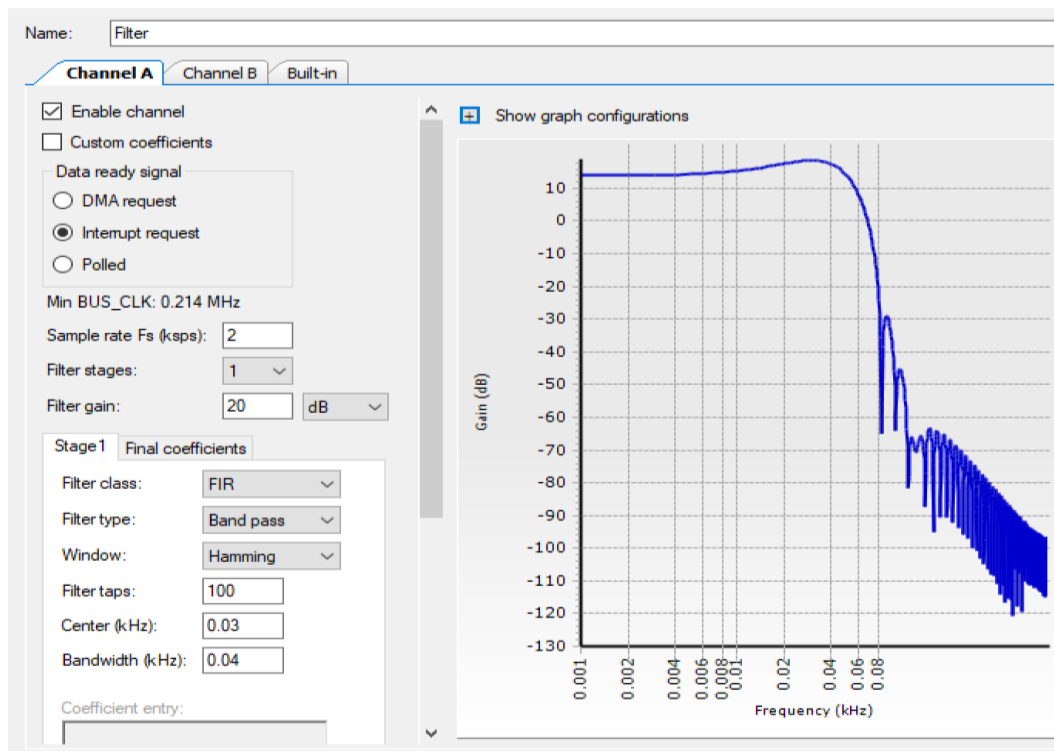


Figura 4.4: Parâmetros do Filtro Passa-Banda.

Após esta fase e como se pretende visualizar em simultâneo, quer o movimento do rato, quer o sinal, recorreu-se a duas vias distintas. Começando pelo eletromiograma, o *interrupt* `isr_Filter` da Figura 4.2, atua como um *interrupt* de hardware. Portanto, cada vez que ocorre um *interrupt request* o sistema é alertado para uma determinada ação que tem de desencadear, como sugere a seguinte função.

```
CY_ISR(Filter_Done)
{
    uint8 vdata;
    uint32 Filter_Result;

    /* Read the filter value from filter's holding register */
    Filter_Result = Filter_Read24(Filter_CHANNEL_A);
}
```



```

        /* Write the MSB eight bits to the VDAC */
        vdata = Filter_Result >> SHIFT_EIGHT;

        VDAC8_SetValue(vdata);

    }

```

Verifica-se que, cada vez que o filtro termina o seu processamento de dados (Filter\_Done), é registado o valor que se encontra no registo A do filtro numa variável Filter\_Result. Dado que o resultado da função Filter\_Read24 é um valor de 32 *bits*, optou-se por trabalhar com os 8 *bits* mais significativos (SHIFT\_EIGHT) deste valor, já que o DAC a usar é também de 8 *bits*. Tendo isto em conta, o bloco VDAC8 recebe o conjunto de data (valores entre 0 e 255) proveniente do filtro e converte de novo num sinal analógico, estando a sua saída associada ao *pin* VDAC, tal como é explícito na Figura 4.2. Como o sinal processado ainda contém *offset* e de forma a remover a componente DC, recorre-se ao filtro passa-alto da secção 3.3, e a saída do filtro fica ligada à entrada do NI myDAQ.

Em relação à configuração do rato e como a saída corresponde à porta Micro-USB, recorre-se a um *loop* infinito, neste caso um ciclo *for*. Em concreto, a cada iteração do ciclo, após a passagem dos dados pelo filtro, desencadeia-se o seguinte código.

```

for(;;)
{
    /* Read the filter value from filter's holding register */
    Filter_Result = Filter_Read24(Filter_CHANNEL_A);

    /* Write the MSB eight bits to the VDAC */
    vdata = Filter_Result >> SHIFT_EIGHT;

    while(!USBFS_1_bGetEPAckState(1)); /* Wait for ACK before loading data */

    USBFS_1_LoadInEP(1, (uint8 *)Mouse_Data, 3); /* Load latest mouse data
    into EP1 and send to PC */

    if(val2 > 230){

        Mouse_Data[1] = -5;          /* Moves mouse to left */
        Mouse_Data[2] = 0;          /* Keeps Y static */

    }

    else if ((val_final>200) && (val_final<210))

    {

        Mouse_Data[1] = 5;           /* Moves mouse to right */
        Mouse_Data[2] = 0;           /* Keeps Y static */

    }
}

```

```

else if ((val_final>180) && (val_final<200))

{
    Mouse_Data[1] = 0;           /* Keeps X static */
    Mouse_Data[2] = 5;           /* Moves mouse down */

}

else if ((val2>210) && (val2<230))

{
    Mouse_Data[1] = 0;           /* Keeps X static */
    Mouse_Data[2] = -5;          /* Moves mouse up */

}

else {
    Mouse_Data[1] = 0;           /* Keeps X static */
    Mouse_Data[2] = 0;           /* Keeps Y static */
}

}

```

Primeiramente, o bloco que permite ao PSoC atuar como um rato para computador é a componente USBFS, visto que pode atuar como um dispositivo do tipo HID (*Human Interface Device*)<sup>18</sup>, tendo sido configurado para atuar como tal. Voltando ao ciclo, repare-se que a parte inicial é semelhante à do *interrupt*: lê-se o valor registado no filtro, para uma variável, e de seguida consideram-se somente os 8 *bits* mais significativos. Quando o bloco USBFS recebe um ACK (*USB Handshake packet indicating a positive acknowledgment*), carrega os dados provenientes do filtro, para o seu *Endpoint* (*A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device*).

De seguida, há um conjunto de condições que vão avaliar o valor recebido e, consoante o mesmo, determinam o movimento do rato.

É de relevar, novamente, que este projeto contém uma componente de tentativa erro muito acentuada e, para além disso, ao longo dos vários meses de trabalho uma das grandes condicionantes, que não foi totalmente ultrapassada, diz respeito à colocação e aderência dos elétrodos à pele, bem como o desgaste dos mesmos por sucessivas utilizações, o que levou a vários resultados díspares nalguns testes. Em

---

<sup>18</sup> Um dispositivo do tipo HID é qualquer dispositivo que permita a interação entre o utilizador e o computador, por exemplo o rato e o teclado.

concreto, para este caso, os valores recebidos entre 0 e 255 por vezes oscilaram, pelo que as condições no código acima dizem respeito aos valores mais comuns registados para cada movimento do rato.

Analisando, comando a comando, e indo ao encontro do sinal que se visualiza no osciloscópio, aquando a movimentação do pulso, percebeu-se que, para a primeira condição:

```
if(val2 > 230){

    Mouse_Data[1] = -5;           /* Moves mouse to left */
    Mouse_Data[2] = 0;           /* Keeps Y static */

}
```

o rato movimenta-se para a esquerda no ecrã, se o valor registado for superior a 230. Tendo em conta que foi definido um vetor `Mouse_Data`, com três posições possíveis - três ações possíveis – em que a posição 0 corresponde ao clique (não implementado), a posição 1 é a do movimento em relação ao eixo do X e a posição 2 a do movimento em relação ao eixo do Y. A Figura 4.5 ilustra o sinal correspondente a esta deslocação do rato, sendo interessante registar o claro efeito do filtro, tal como esperado.

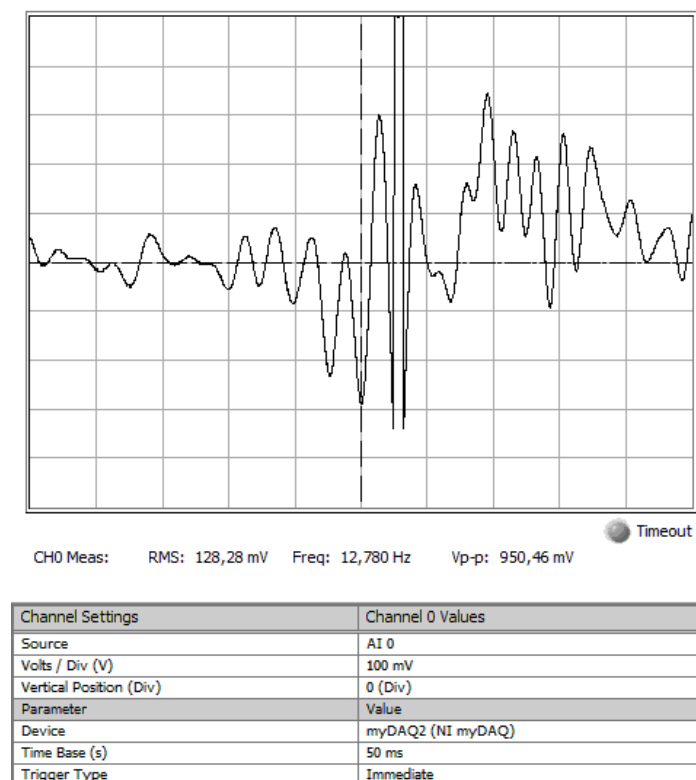


Figura 4.5: Eletromiograma do Comando Esquerda.

Para os outros *if*, do ciclo *for*, o funcionamento é inteiramente semelhante (cada situação é explicada com comentários nas linhas de código), sendo de salientar os diferentes sinais que se registam, para estes casos na Figura 4.6, Figura 4.7, Figura 4.8 e Figura 4.9. Recuperando a secção 3.4, em que foi possível entender que se pode definir diferentes níveis de força, todos os sinais destas figuras estão alinhados com isso, pois denotam valores de tensão pico a pico distintos, o que irá permitir registar valores diferentes para as condições acima mencionadas. Desta forma, entende-se que os comandos da Figura 4.1, permitem ao músculo flexor contrair mais ou menos e por isso gerar sinais de maior, ou menor amplitude.

O restante código desta implementação encontra-se em apêndice, sendo que consiste essencialmente na declaração de funções de inicialização das componentes da Figura 4.2.

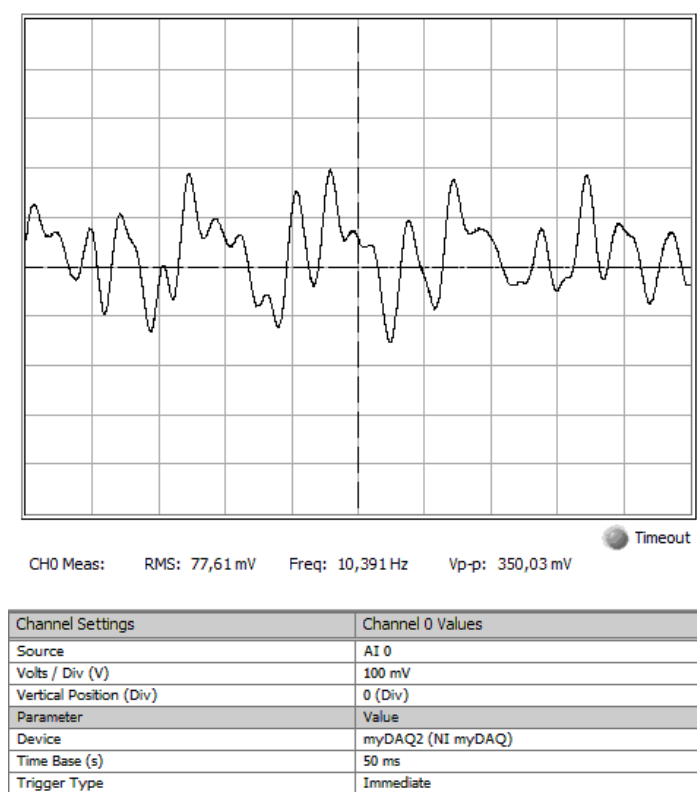


Figura 4.6: Eletromiograma do Comando Direita.

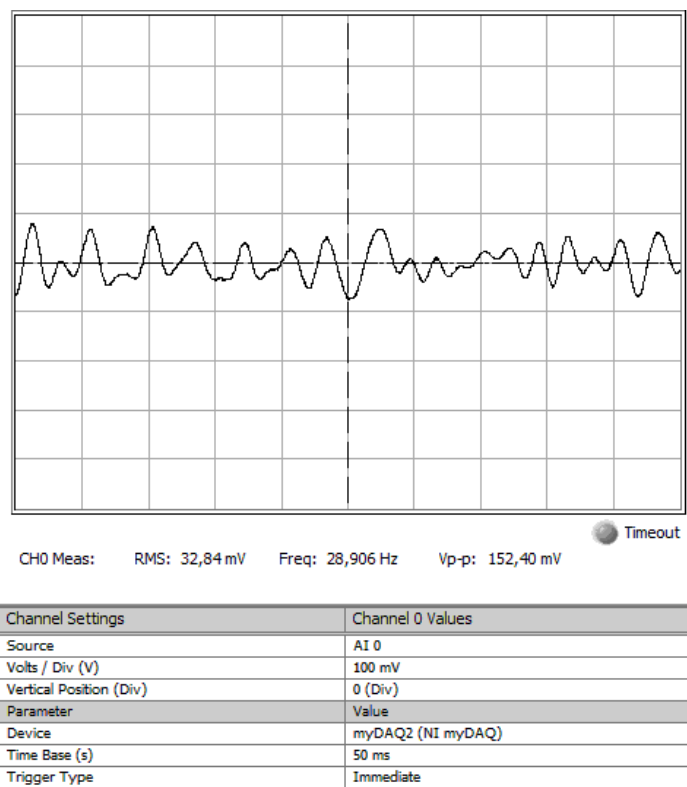


Figura 4.7: Eletromiograma do Comando Baixo.

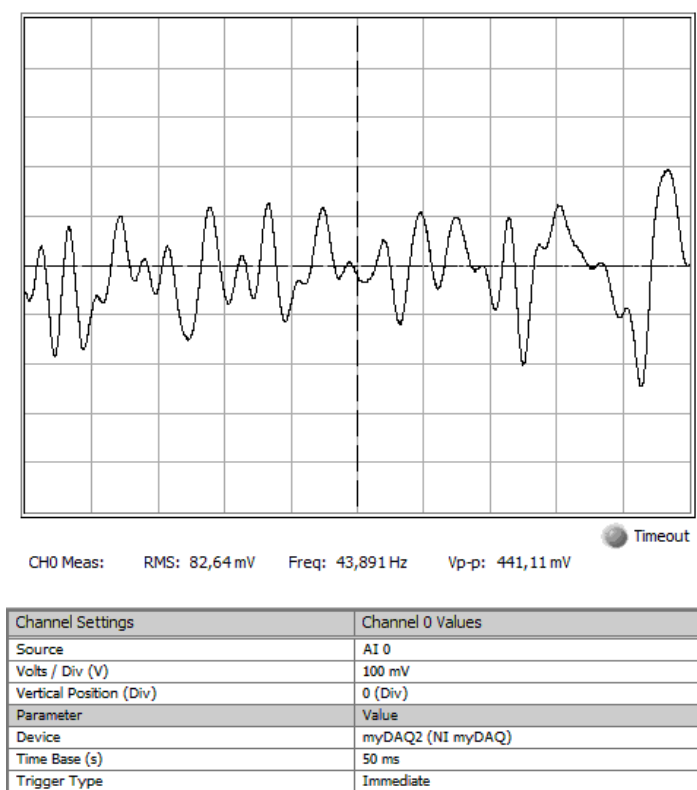


Figura 4.8: Eletromiograma do Comando Cima.

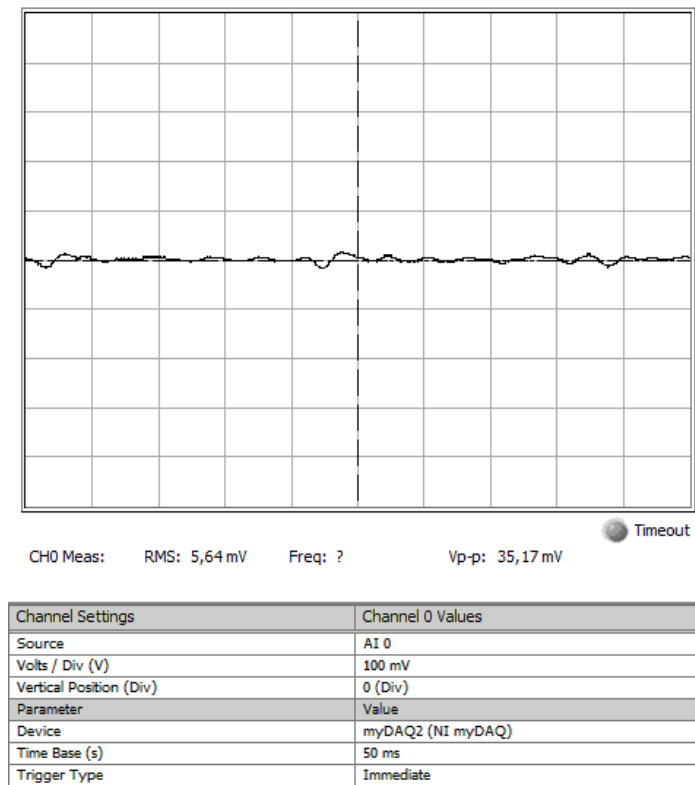


Figura 4.9: Eletromiograma Sem Comando.

O objetivo do trabalho foi atingido. Porém, o comportamento do rato tem algumas falhas, provavelmente não só devido a problemas de aderência dos elétrodos à pele, mas, também, atentando às características do eletromiograma. Pelas figuras apresentadas anteriormente, verifica-se que o sinal tem bastantes oscilações de amplitude e, pelas condições acima mencionados, o objetivo passa por detetar o pico da onda. Concretamente, o valor em que o sinal atinge a sua máxima amplitude, na sequência de cada contração do músculo, o que levará a um determinado valor entre 0 e 255. Para resolver esta questão, recorreu-se a um detetor de pico, que se traduz no circuito da Figura 4.10.

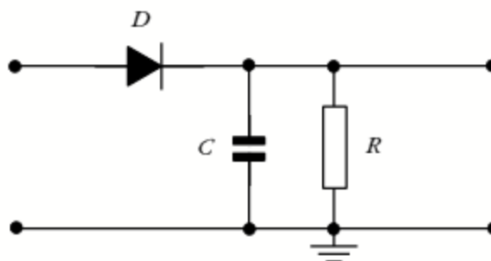


Figura 4.10: Detetor de pico (Adaptado de [21]).

O comportamento desta montagem é bastante simples, o díodo vai funcionar como interruptor. Ou seja, permite a passagem de corrente, quando a tensão de entrada é superior à tensão do condensador, carregando o mesmo com essa tensão mas bloqueando qualquer corrente de fluir de volta, quando a amplitude de entrada está abaixo da tensão do condensador. Neste caso, o condensador é descarregado pela resistência. Assim sendo, quando se tem um sinal com uma determinada frequência e um detetor de pico com um a constante de tempo,  $\tau=RC$ , o tempo entre cada pico será dado pela expressão (4.1).

$$T = \frac{1}{f} \quad (4.1)$$

Considerando  $V_{pico}$  a tensão que carrega o condensador, a expressão (4.2) diz respeito à tensão resultante ( $V'_{pico}$ ), após o condensador ser descarregado.

$$V'_{pico} = V_{pico} \times e^{-\frac{T}{\tau}} \quad (4.2)$$

Exemplificando com uma senoide (onda de entrada), o detetor de pico terá o comportamento da Figura 4.11. Note-se que, para a onda de saída há uma diminuição do valor da tensão quando esta está abaixo da tensão de pico (descarga do condensador), atingindo novamente o valor máximo quando deteta outro pico de tensão.



Figura 4.11: Sinusoide e detetor de pico (Adaptado de [22]).

O eletromiograma não sendo uma senoide acaba por ser um sinal em que é possível aplicar este tipo de circuito. Como o processamento do eletromiograma é feito no *kit* PSoC, é possível desempenhar esta função sem recorrer à montagem da Figura 4.10. Para tal, tanto o código do ciclo *for* como o do *interrupt*, sofre uma ligeira alteração, sendo adicionada uma condição.

```

if (val_final > vdata)

    val_final = val_final*0.9999;

else

    val_final = vdata;

```

Definindo vdata como o valor registado no filtro e val\_final inicialmente 0, quando val\_final for superior a vdata, que acontece após um pico, tem-se o equivalente à expressão (4.2), existindo um determinado valor ajustável, neste caso, por ensaios, entendeu-se que seria 0,9999, que atua como  $e^{-\frac{T}{\tau}}$ . Caso contrário, sempre que val\_final seja menor do que vdata, val\_final assume o valor de vdata. O resultado, desta alteração no código, pode ser visto, num dos quatro casos da contração do flexor, na Figura 4.12.

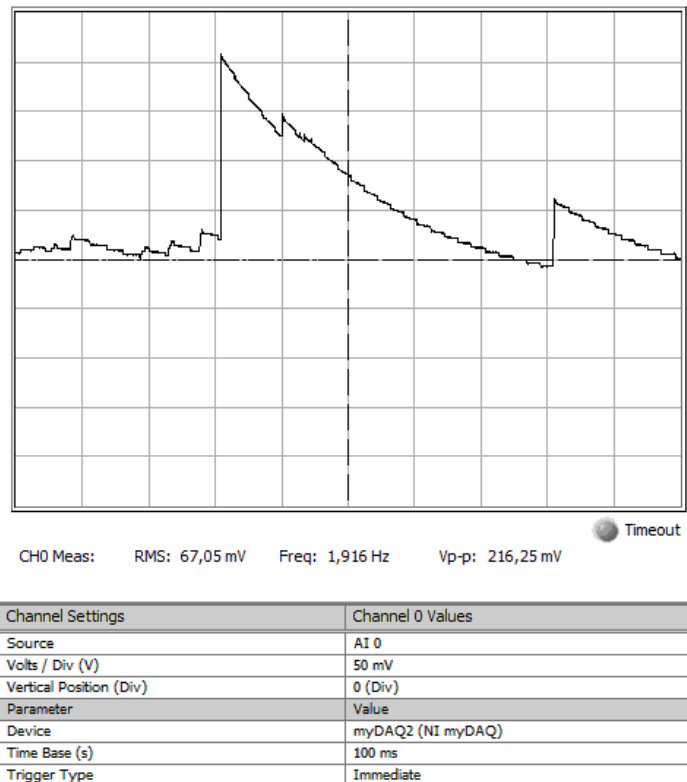


Figura 4.12: Contração do Flexor - Detetor de Pico.

A Figura 4.13 agrega quatro gráficos, já apresentados anteriormente, e que resumem, por completo, toda a evolução no tratamento do sinal, desde que é captado, amplificado, filtrado e após a passagem pelo detetor de pico.



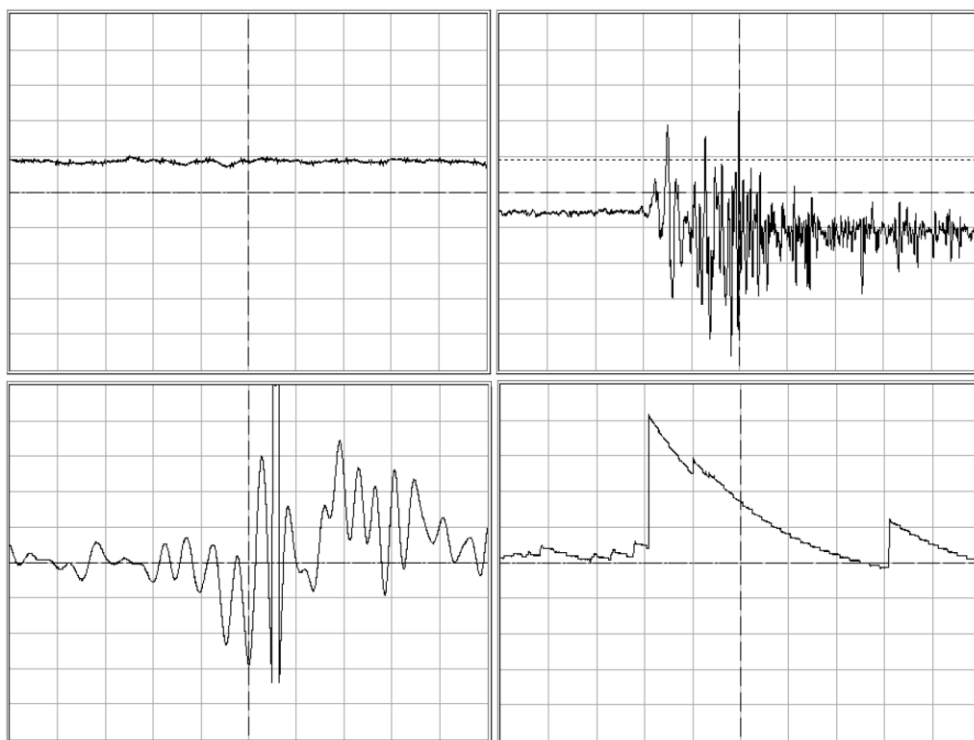


Figura 4.13: Evolução do sinal desde a captação ao processamento - a) Sinal sem amplificação; b) Sinal amplificado; c) Sinal filtrado; d) Sinal após detetor de pico.

# 5

## Conclusões e Trabalho Futuro

### 5.1. Conclusões

De um ponto de vista global, os objetivos deste trabalho foram atingidos, sendo importante retirar conclusões de cada etapa deste projeto, quer no que diz respeito aos pontos positivos, como a aprendizagem adquirida e a resultados de acordo com o esperado, quer no que diz respeito a pontos negativos, em que certos detalhes só parcialmente foram concluídos com sucesso ou, até mesmo, apresentaram resultados inconclusivos.

O Capítulo 2, referente ao estado da arte, pretendeu essencialmente fazer um resumo da bibliografia existente, referente aos diversos métodos, técnicas, tecnologias e, também, juntando estas três componentes, referir projetos já realizados nesta área, o que permitiu retirar algumas ideias relevantes para o desenvolvimento desta dissertação. Numa primeira análise, entendeu-se que seria importante perceber quais os sinais biomédicos mais relevantes e quais as suas características, tendo sempre como foco relevar o porquê da escolha do eletromiograma, em detrimento de outros tipos destes sinais. De seguida, há uma primeira abordagem em relação aos elétrodos, tendo sido feito um estudo do tipo de elétrodos existentes adequados a eletromiografias e, talvez o mais relevante desta secção, perceber quais os tipos de ruído associados, aquando da realização de uma eletromiografia, o que conduziu a um estudo da melhor colocação possível do elétrodo no músculo.

Segue-se o estudo do condicionamento do sinal, no que diz respeito aos amplificadores e filtros e, juntando essas duas componentes, uma análise de dois modelos aplicados em projetos distintos, que se mostraram bastante eficientes para conseguir visualizar, com mais precisão, o sinal mioelétrico.

A aquisição de dados, referente ao processamento do sinal, foi o tema seguinte, tendo sido dado grande relevo ao ADC e à sua aplicação nos dispositivos de aquisição de dados, de que são exemplo a placa de som e o DAQ. É ainda analisado, com detalhe, o modelo computacional RNA, referente à classificação do EMG, abordando a componente mais simples de uma rede neuronal e evoluindo para modelos mais avançados e aplicados em vários casos de estudo com taxas de sucesso consideráveis.

Em termos de modelos de aplicação, excluindo o rato para computador, que é o objetivo desta tese, foram analisados alguns protótipos que recorrem a sinais musculares e que vieram contribuir para um aumento da qualidade de vida dos respetivos utilizadores, demonstrando a pertinência deste tipo de projetos. Resumidamente, o Capítulo 2 referencia a aquisição de todo o conhecimento teórico necessário para colocar em prática a ideia deste trabalho, constituindo, por isso, a base do racional do tema e desenvolvimento desta tese.

O Capítulo 3, foi, sem dúvida, a etapa de maior relevância, pelo confronto com o “pôr em prática”, que, inevitavelmente, trouxe mais avanços em termos de aprendizagem do sinal do eletromiograma e das escolhas tecnológicas e, ao mesmo tempo, trouxe as complicações da respetiva visualização e da concretização prática do objetivo. Os diversos testes efetuados neste capítulo e os respetivos resultados obtidos foram divididos por fases, desde a captação do sinal até ao processamento do mesmo. As conclusões a retirar são determinadas pelas situações de tentativa/erro, mais do que pela análise de cada uma das secções deste capítulo, daí as referências anteriores a este tipo de constrangimentos.

Com convicção, a captação do sinal é o principal fator a realçar neste capítulo, visto que houve grande dificuldade em escolher os melhores elétrodos a utilizar, embora tenham sido definidos dois ou três modelos, como sendo os melhores, até por sugestão de pessoas qualificadas para o efeito. Na verdade, mesmo assim, foram necessárias muitas horas de testes para conseguir resultados positivos, sobretudo porque esses resultados dependiam, como anteriormente referido, da aderência dos elétrodos à pele e do desgaste dos mesmos, influenciando, conseqüentemente, o desempenho do rato. Há a realçar, como aspeto positivo, que tanto o dispositivo de aquisição de dados, como a placa de amplificação demonstraram ser bastante eficientes, contribuindo, significativamente, para conseguir resultados relevantes para o projeto. O programa Matlab, juntamente com o PSoC, foram ferramentas essenciais para conseguir perceber a atuação do filtro no sinal eletromiográfico.

Por último, o Capítulo 4, que agrega todo este conhecimento prévio teórico-prático, de forma a concretizar o objetivo deste trabalho de implementar o movimento do rato. Para o efeito, foi desenvolvido um modelo simples e recorreu-se, essencialmente, ao *kit* PSoC, no que diz respeito à programação do periférico de entrada em questão. Em concreto, há uma abordagem semelhante, aos modelos analisados no Capítulo 2, com amplificação e filtragem do EMG mas não se recorreu, por exemplo, a uma rede neuronal, optando-se por um conjunto de critérios e condições, para avaliar os

valores enviados, após a atuação do filtro e, posteriormente, a atuação de um detetor de pico, para melhorar o desempenho do rato, eliminando as constantes variações de amplitude da onda.

A Figura 5.1 ilustra a montagem referente à implementação final, no momento da contração do flexor para a esquerda.

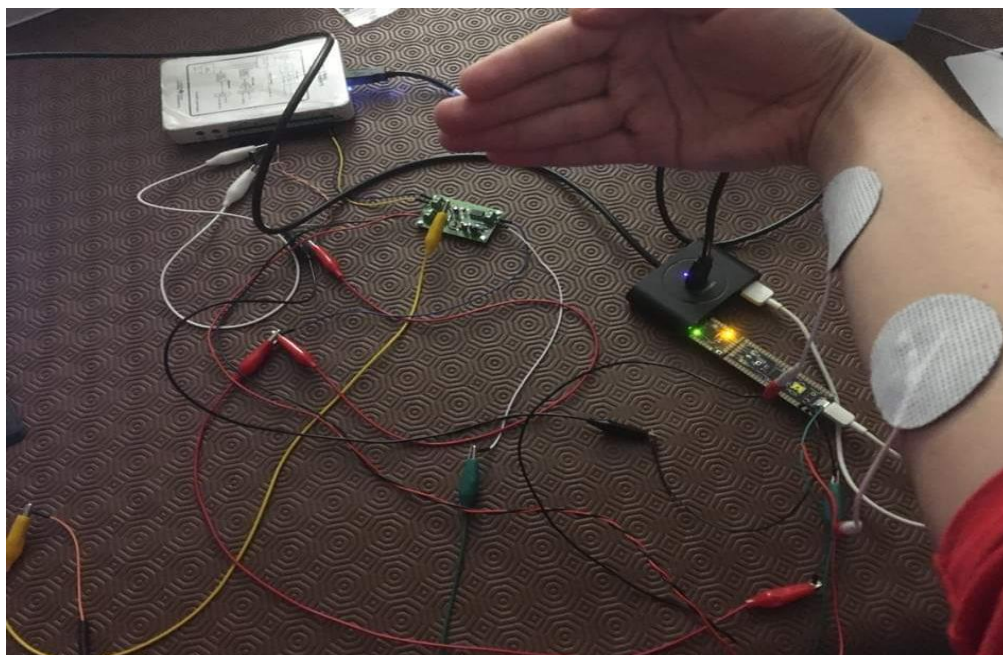


Figura 5.1: Montagem da Implementação Final.

## 5.2. Trabalho Futuro

Há diversos aspetos que poderão ser melhorados e outras componentes ser acrescentadas ao protótipo. Em primeiro lugar, o destaque vai para a necessidade de melhorar o desempenho da aquisição do sinal, de forma a minimizar, idealmente eliminar, a tentativa/erro para conseguir captar o EMG nas melhores condições. Admite-se que a solução possa passar pela utilização de outro tipo de elétrodos, com uma colocação mais precisa e de melhor aderência à pele. Por outro lado, existirão, possivelmente, métodos mais eficientes em relação ao condicionamento do sinal e no que diz respeito ao processamento do mesmo. Mas, em relação a essa parte, as técnicas utilizadas revelaram-se bastante satisfatórias.

A acrescentar, seria interessante desenvolver mais o rato, nomeadamente, com a concretização do clique (*enter*), em complemento dos quatro comandos aqui resolvidos e tornando a solução totalmente funcional para o utilizador. Para a eficiência ainda ser maior, desenvolver um algoritmo, eventualmente com recurso a redes neuronais, que permita movimentar o rato em todas as direções possíveis. Para que isto seja possível, tendo em conta múltiplas variáveis associadas, seria relevante desenvolver um sistema de calibração que adaptaria previamente o rato ao utilizador, i.e., a pessoa treinava o sistema, realizando

os vários comandos possíveis. Assim, o reconhecimento das características do sinal desse utilizador faria com que o rato muito provavelmente correspondesse a 100% aos movimentos do pulso efetuados.

## Bibliografia

- [1] «Data Acquisition System Development with MATLAB». [Em linha]. Disponível em: <https://www.mathworks.com/discovery/data-acquisition-system.html>. [Acedido: 05-Set-2018].
- [2] R. M. Rangayyan, Ed., «Introduction to Biomedical Signals», em *Biomedical Signal Analysis*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015, pp. 1–69.
- [3] F. Sadikoglu, C. Kavalcioglu, e B. Dagman, «Electromyogram (EMG) signal detection, classification of EMG signals and diagnosis of neuropathy muscle disease», *Procedia Computer Science*, vol. 120, pp. 422–429, 2017.
- [4] P. E. Barkhaus e S. D. Nandedkar, «Recording characteristics of the surface EMG electrodes», *Muscle & Nerve*, vol. 17, n. 11, pp. 1317–1323, Nov. 1994.
- [5] A. V. Vidyapeetham, «AMRITA VISHWA VIDYAPEETHAM COIMBATORE 641 105 APRIL 2010», p. 91.
- [6] «Ambu® Neuroline Inoject - Électrode emg / aiguille by Ambu | MedicalExpo». [Em linha]. Disponível em: <http://www.medicalexpo.fr/prod/ambu/product-69019-782908.html>. [Acedido: 05-Set-2018].
- [7] «The Instrumentation Amplifier | Operational Amplifiers | Electronics Textbook». [Em linha]. Disponível em: <https://www.allaboutcircuits.com/textbook/semiconductors/chpt-8/the-instrumentation-amplifier/>. [Acedido: 05-Set-2018].
- [8] «Chegg.com». [Em linha]. Disponível em: <https://www.chegg.com/homework-help/questions-and-answers/design-passive-first-order-rc-low-pass-filter-cutoff-frequency-fh-10khz-derive-step-respon-q1947105>. [Acedido: 05-Set-2018].
- [9] «Chegg.com». [Em linha]. Disponível em: <https://www.chegg.com/homework-help/questions-and-answers/rc-high-pass-filter-shown-figure-r-155-k-c-00550-f-3-db-frequency-circuit-frequency-ratio-q2428509>. [Acedido: 05-Set-2018].
- [10] A. Pradhan *et al.*, «Acquisition and classification of EMG using a dual-channel EMG biopotential amplifier for controlling assistive devices», em *2016 IEEE Annual India Conference (INDICON)*, Bangalore, India, 2016, pp. 1–5.
- [11] J. Wang, L. Tang, e J. E Bronlund, «Surface EMG Signal Amplification and Filtering», *International Journal of Computer Applications*, vol. 82, n. 1, pp. 15–22, Nov. 2013.
- [12] «Lynguent Part I - November 15, 2010», *EDACafe*. [Em linha]. Disponível em: <https://www10.edacafe.com/nbc/articles/2/879600/Lynguent-Part>. [Acedido: 05-Set-2018].
- [13] «Perceptron». [Em linha]. Disponível em: [http://www.saedsayad.com/artificial\\_neural\\_network\\_bkp.htm](http://www.saedsayad.com/artificial_neural_network_bkp.htm). [Acedido: 05-Set-2018].

- [14] R. Ahsan, M. I. Ibrahimy, e O. O. Khalifa, «EMG Signal Classification for Human Computer Interaction: A Review», p. 23.
- [15] M. B. I. Reaz, M. S. Hussain, e F. Mohd-Yasin, «Techniques of EMG signal analysis: detection, processing, classification and applications», *Biological Procedures Online*, vol. 8, n. 1, pp. 11–35, Dez. 2006.
- [16] «Hand of Hope: EMG-driven Robotic Arm for After Stroke Rehab - Cool Wearable». [Em linha]. Disponível em: <http://www.coolwearable.com/hand-of-hope/>. [Acedido: 05-Set-2018].
- [17] «NI myDAQ User Guide - National Instruments», p. 40.
- [18] A. Devices, «Low Cost Low Power Instrumentation Amplifier AD620», p. 20.
- [19] «Muscles of the Upper Limb», p. 6, Ago. 2012.
- [20] «Low-cost embedded development kits accelerate design cycles». [Em linha]. Disponível em: <http://www.embedded-computing.com/embedded-computing-design/low-cost-embedded-development-kits-accelerate-design-cycles>. [Acedido: 01-Fev-2019].
- [21] «The Envelope Detector». [Em linha]. Disponível em: [https://www.st-andrews.ac.uk/~www\\_pa/Scots\\_Guide/RadCom/part9/page2.html](https://www.st-andrews.ac.uk/~www_pa/Scots_Guide/RadCom/part9/page2.html). [Acedido: 27-Fev-2019].
- [22] B. Linares-Barranco, T. Serrano-Gotarredona, J. Ramos-Martos, J. Ceballos-Caceres, J. M. Mora, e A. Linares-Barranco, «A precise 90/spl deg/ quadrature OTA-C oscillator tunable in the 50-130-MHz range», *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, n. 4, pp. 649–663, Abr. 2004.

# Apêndices

## 1. Código referente à Figura 3.24

```
#include <project.h>

#define DEVICE_0          0
#define USB_BUFFER_SIZE   62
#define USBUART_EndPoint_Number 0x02

/* DMA Configuration for DMA */
#define DMA_BYTES_PER_BURST 2
#define DMA_REQUEST_PER_BURST 1
#define DMA_SRC_BASE (CYDEV_PERIPH_BASE)
#define DMA_DST_BASE (CYDEV_SRAM_BASE)

void DMA_Config(void);

/* Variable declarations for DMA */
uint8 DMA_Chain;
uint8 DMA_TD[1];

uint8 DMA_done = 0;

uint8 ADC_sample[USB_BUFFER_SIZE] = {0};

int8 val;

int main()
{
    DMA_Config();

    /* Enable global interrupt */
    CYGlobalIntEnable;

    /* Start USBFS operation with 3V operation */
    USBUART_Start(DEVICE_0, USBUART_5V_OPERATION);

    /* Wait for device to enumerate */
    while(!USBUART_GetConfiguration());

    /* Initialize USBUART */
    USBUART_CDC_Init();

    /* Setup the USB component for automatic DMA transfers */
    USBUART_LoadInEP(USBUART_EndPoint_Number , ADC_sample, 62);

    /* Start ADC */
```



```

ADC_DelSig_Start();

/* Disable the ADC ISR as it is not required */
ADC_DelSig_IRQ_Enable();

/* Start the interrupt to identify end of DMA transfers */
DMA_Done_isr_Start();

/* Start ADC conversion */
ADC_DelSig_StartConvert();

for(;;)
{
    if(DMA_done)
    {
        /* Stop ADC Conversion for transferring the buffered data
*/
        ADC_DelSig_StopConvert();

        /* Check for USB ready. */
        if(USBUART_CDCIsReady() != 0u)
        {
            /* Transfer data - can only send 31 ADC results (64
bytes)
            * because UART needs null termination. */
            USBUART_LoadInEP(USBUART_EndPoint_Number, 0 , 62);

            /* Reset the flag set in DMA Done ISR */
            DMA_done = 0;

            /* Delay to allow for human validation of
transferred data.
            * This is not required in a real project. */
            //CyDelay(1);

            /* Restart ADC conversions */
            ADC_DelSig_StartConvert();
        }
    }
} /* for loop ends here */
}

void DMA_Config(void) {
    /* DMA Initialization - 2 byte per trigger. Each burst transfer
    * needs a new request. Set upper source and destination address. */

    DMA_Chanel = DMA_DmaInitialize(DMA_BYTES_PER_BURST,
DMA_REQUEST_PER_BURST,
                                HI16(DMA_SRC_BASE), HI16(DMA_DST_BASE));

    /* Allocate TD */
    DMA_TD[0] = CyDmaTdAllocate();

    /* Destination address is incremented after each transaction and

```

```

        * Term out generated after the specified no. of bytes are
transferred
        * to memory. Transfer the specified number of samples and generate
        * term out */
CyDmaTdSetConfiguration(DMA_TD[0], 62, DMA_TD[0],
                        DMA__TD_TERMOUT_EN | TD_INC_DST_ADR);
#endif

/* Set source and destination address */
CyDmaTdSetAddress(DMA_TD[0], L016((uint32)ADC_DeISig_DEC_SAMP_PTR),
                  L016((uint32)ADC_sample));

/* Set the intial TD for the channel */
CyDmaChSetInitialTd(DMA_Chan, DMA_TD[0]);

/* Enable the DMA channel */
CyDmaChEnable(DMA_Chan, 1);
}

```

## 2. Código referente à secção 3.6

```

np=length(input1)-1;
Ts=1/1953; %frequencia de amostragem 1953 Hz
time=0:Ts:np*Ts;

figure (1)

xlabel('Tempo (s)');
ylabel('Amplitude');
hold on
plot(time,input1)

figure(2)

xlabel('Frequência (Hz)');
ylabel('Amplitude');
hold on

yf=fft(input1);
ff=0:1/(Ts*np):1/Ts;

plot(ff,20*log10(abs(yf)))

figure (3)

Wn=[10 50]/(1953/2)

a = fir1(100,Wn, 'bandpass')

freqz(a,1)

figure (4)

y2 = filter(a,1,input1)

```

```

xlabel('Tempo (s)');
ylabel('Amplitude');
hold on

plot (time,y2)

```

### 3. Código referente à Figura 4.2

```

#include <device.h>

#include <project.h>

#include <math.h>

/* Necessary defines for DMA Configuration. Request Per Burst is set to
1 as ADC
End Of Conversion triggers DMA. Two Bytes are transferred per burst to
write into
lower and middle Staging Registers of the filter block */

#define DMA_BYTES_PER_BURST      (2u)
#define DMA_REQUEST_PER_BURST   (1u)
#define DMA_SRC_BASE              (CYDEV_PERIPH_BASE)
#define DMA_DST_BASE              (CYDEV_PERIPH_BASE)
#define TRANSFER_COUNT           (2u)

/* General constat defines */
#define FILTER_DATA_ALIGN        (0x05u)
#define SHIFT_EIGHT              (0x08u)
#define RESCALING_FACTOR         (0x80u)

/* Function Prototypes */
/* Interrpt service routine to read the filterd data */
CY_ISR_PROTO(Filter_Done);

/* Function to configure DMA Channel */
void DMA_Config(void);

static int8 Mouse_Data[3] = {0, 0, 0}; /* [0] = Buttons, [1] = X-Axis,
[2] = Y-Axis */
static uint16 Delay = 0;

float32 val1 = 0;

float32 val2 = 0;

float32 val3 = 0;

void main()
{
    uint8 vdata;
    uint32 Filter_Result;
    /* Enable Global Interrupts */
    CYGlobalIntEnable;

```

```

    /* Start all components used on schematic */
    ADC_DelSig_Start();
    Filter_Start();
    VDAC8_Start();
    ADC_DelSig_IRQ_Start();
    isr_Filter_StartEx(Filter_Done);

    /* For 9-16 bits filter resolution, coherency should be mid, Dalign
    should be enabled,
    filter stage pointer will be Filter_STAGE_A_PTR */
    Filter_DALIGN_REG = Filter_DALIGN_REG | FILTER_DATA_ALIGN;
    Filter_SetCoherency(Filter_CHANNEL_A, Filter_KEY_MID );

    /* User-implemented function to set-up DMA */
    DMA_Config();

    /* Start the ADC Conversion */
    ADC_DelSig_StartConvert();

    USBFS_1_Start(0, USBFS_1_DWR_VDDD_OPERATION); /* Start USBFS
    Operation/device 0 and with 5V operation */
    while(!USBFS_1_bGetConfiguration()); /* Wait for
    Device to enumerate */
    USBFS_1_LoadInEP(1, (uint8 *)Mouse_Data, 3); /* Loads an initial
    value into EP1 and sends it out to the PC */

    for(;;)
    {

        /* Read the filter value from filter's holding register */
        Filter_Result = Filter_Read24(Filter_CHANNEL_A);

        /* Write the MSB eight bits to the VDAC */
        vdata = Filter_Result >> SHIFT_EIGHT;

        while(!USBFS_1_bGetEPAckState(1)); /* Wait for ACK before loading data */

        USBFS_1_LoadInEP(1, (uint8 *)Mouse_Data, 3); /* Load latest mouse data
        into EP1 and send to PC */

        if(val2 > 230){

            Mouse_Data[1] = -5; /* Moves mouse to left */
            Mouse_Data[2] = 0; /* Keeps Y static */

        }

        else if ((val_final>200) && (val_final<210))

        {

            Mouse_Data[1] = 5; /* Moves mouse to right */
            Mouse_Data[2] = 0; /* Keeps Y static */

        }

        else if ((val_final>180) && (val_final<200))

```

```

    {
        Mouse_Data[1] = 0;          /* Keeps X static */
        Mouse_Data[2] = 5;          /* Moves mouse down */
    }

    else if ((val2>210) && (val2<230))

    {
        Mouse_Data[1] = 0;          /* Keeps X static */
        Mouse_Data[2] = -5;         /* Moves mouse up */
    }

    else {
        Mouse_Data[1] = 0;          /* Keeps X static */
        Mouse_Data[2] = 0;          /* Keeps Y static */
    }

}

CY_ISR(Filter_Done)
{
    uint8 vdata;
    uint32 Filter_Result;

    /* Read the filter value from filter's holding register */
    Filter_Result = Filter_Read24(Filter_CHANNEL_A);

    /* Write the MSB eight bits to the VDAC */
    vdata = Filter_Result >> SHIFT_EIGHT;

    /* Add 128 to convert 8 bit signed number to 8 bit unsigned number
*/
    VDAC8_SetValue(vdata + RESCALING_FACTOR);

}
/*****
*****
* Function Name: DMA_Config
*****
*****
*
* Summary:
*   Initializes and sets up DMA for use (generated by DMA Wizard)
*
* Parameters:
*   None.
*
* Return:
*   None.
*
*/

```

```

*****
*****/

void DMA_Config(void)
{
    /* Variable declarations for DMA */
    uint8 channelHandle;
    uint8 tdChanA;

    /* Configure the DMA to Transfer the data in 1 burst with individual
    trigger
    * for each burst.*/
    channelHandle = DMA_DmaInitialize(DMA_BYTES_PER_BURST,
    DMA_REQUEST_PER_BURST,
                                HI16(DMA_SRC_BASE),
    HI16(DMA_DST_BASE));

    /* This function allocates a TD for use with an initialized DMA
    channel */
    tdChanA = CyDmaTdAllocate();

    /* Source and Destination address increments are not needed as we
    are using 2
    byte transfers and Spoke Width is 16 bit */
    CyDmaTdSetConfiguration(tdChanA, TRANSFER_COUNT, tdChanA, 0u);

    /* Set the source address as ADC_DeISig and the destination as
    Filter Channel A*/
    CyDmaTdSetAddress(tdChanA, LO16((uint32)ADC_DeISig_DEC_SAMP_16B_PTR),
    LO16((uint32)Filter_STAGE_A_PTR));

    /* Set tdChanA to be the initial TD associated with channelHandle */
    CyDmaChSetInitialTd(channelHandle, tdChanA);

    /* Enable the DMA channel represented by channelHandle and preserve
    the TD */
    CyDmaChEnable(channelHandle, 1u);
}

/* [] END OF FILE */

/* End of File */

```

#### 4. Código referente à Figura 4.2, com detetor de pico

```

#include <device.h>

#include <project.h>

#include <math.h>

/* Necessary defines for DMA Configuration. Request Per Burst is set to 1
as ADC
End Of Conversion triggers DMA. Two Bytes are transferred per burst to
write into
lower and middle Staging Registers of the filter block */

#define DMA_BYTES_PER_BURST      (2u)

```

```

#define DMA_REQUEST_PER_BURST    (1u)
#define DMA_SRC_BASE              (CYDEV_PERIPH_BASE)
#define DMA_DST_BASE              (CYDEV_PERIPH_BASE)
#define TRANSFER_COUNT            (2u)

/* General constat defines */
#define FILTER_DATA_ALIGN         (0x05u)
#define SHIFT_EIGHT              (0x08u)
#define RESCALING_FACTOR          (0x80u)

/* Function Prototypes */
/* Interrpt service routine to read the filterd data */
CY_ISR_PROTO(Filter_Done);

/* Function to configure DMA Channel */
void DMA_Config(void);

static uint8 Mouse_Data[3] = {0, 0, 0}; /* [0] = Buttons, [1] = X-Axis, [2]
= Y-Axis */
static uint16 Delay = 0;

float32 val1 = 0;

float32 val2 = 0;

float32 val_final = 0;

void main()
{
    uint8 vdata;
    uint32 Filter_Result;
    /* Enable Global Interrupts */
    CYGlobalIntEnable;

    /* Start all components used on schematic */
    ADC_DelSig_Start();
    Filter_Start();
    VDAC8_Start();
    ADC_DelSig_IRQ_Start();
    isr_Filter_StartEx(Filter_Done);

    /* For 9-16 bits filter resolution, coherency should be mid, Dalign
should be enabled,
filter stage pointer will be Filter_STAGEA_PTR */
    Filter_DALIGN_REG = Filter_DALIGN_REG | FILTER_DATA_ALIGN;
    Filter_SetCoherency(Filter_CHANNEL_A, Filter_KEY_MID );

    /* User-implemented function to set-up DMA */
    DMA_Config();

    /* Start the ADC Conversion */
    ADC_DelSig_StartConvert();

    USBFS_1_Start(0, USBFS_1_DWR_VDDD_OPERATION); /* Start USBFS
Operation/device 0 and with 5V operation */
    while(!USBFS_1_bGetConfiguration()); /* Wait for
Device to enumerate */

```

```

    USBFS_1_LoadInEP(1, (uint8 *)Mouse_Data, 3);    /* Loads an initial
value into EP1 and sends it out to the PC */

    for(;;)
    {
        /* Read the filter value from filter's holding register */
        Filter_Result = Filter_Read24(Filter_CHANNEL_A);

        /* Write the MSB eight bits to the VDAC */
        vdata = Filter_Result >> SHIFT_EIGHT;

        if (val_final > vdata)

            val_final = val_final*0.9999;

        else

            val_final = vdata;

        while(!USBFS_1_bGetEPAckState(1));    /* Wait for ACK before
loading data */

        USBFS_1_LoadInEP(1, (uint8 *)Mouse_Data, 3); /* Load latest mouse
data into EP1 and send to PC */

        if(val_final > 230){

            Mouse_Data[1] = -5;    /* Moves mouse to left */
            Mouse_Data[2] = 0;    /* Keeps Y static */

        }

        else if ((val_final >200) && (val_final <210))
        {

            Mouse_Data[1] = 5;    /* Moves mouse to right */
            Mouse_Data[2] = 0;    /* Keeps Y static */

        }

        else if ((val_final >180) && (val_final <200))

        {

            Mouse_Data[1] = 0;    /* Keeps X static */
            Mouse_Data[2] = 5;    /* Moves mouse down */

        }
    }

```



```

else if ((val_final >210) && (val_final <230))

{
    Mouse_Data[1] = 0;           /* Keeps X static */
    Mouse_Data[2] = -5;         /* Moves mouse up */

}

else {
    Mouse_Data[1] = 0;           /* Keeps X static */
    Mouse_Data[2] = 0;           /* Keeps Y static */
}
}

CY_ISR(Filter_Done)
{
    uint8 vdata;
    uint32 Filter_Result;

    /* Read the filter value from filter's holding register */
    Filter_Result = Filter_Read24(Filter_CHANNEL_A);

    /* Write the MSB eight bits to the VDAC */
    vdata = Filter_Result >> SHIFT_EIGHT;

    if (val_final > vdata)

        val_final = val_final*0.9999;

    else

        val_final = vdata;

    /* Add 128 to convert 8 bit signed number to 8 bit unsigned number */
    VDAC8_SetValue(val_final + RESCALING_FACTOR);
}

/*****
* Function Name: DMA_Config
*****/
*
* Summary:
*   Initializes and sets up DMA for use (generated by DMA Wizard)
*
* Parameters:
*   None.
*
* Return:
*   None.
*

```

```

*****
****/
void DMA_Config(void)
{
    /* Variable declarations for DMA */
    uint8 channelHandle;
    uint8 tdChanA;

    /* Configure the DMA to Transfer the data in 1 burst with individual
trigger
    * for each burst.*/
    channelHandle = DMA_DmaInitialize(DMA_BYTES_PER_BURST,
DMA_REQUEST_PER_BURST,
                                     HI16(DMA_SRC_BASE),
HI16(DMA_DST_BASE));

    /* This function allocates a TD for use with an initialized DMA channel
*/
    tdChanA = CyDmaTdAllocate();

    /* Source and Destination address increments are not needed as we are
using 2
byte transfers and Spoke Width is 16 bit */
    CyDmaTdSetConfiguration(tdChanA, TRANSFER_COUNT, tdChanA, 0u);

    /* Set the source address as ADC_DeISig and the destination as Filter
Channel A*/
    CyDmaTdSetAddress(tdChanA, LO16((uint32)ADC_DeISig_DEC_SAMP_16B_PTR),
LO16((uint32)Filter_STAGE_A_PTR));

    /* Set tdChanA to be the initial TD associated with channelHandle */
    CyDmaChSetInitialTd(channelHandle, tdChanA);

    /* Enable the DMA channel represented by channelHandle and preserve the
TD */
    CyDmaChEnable(channelHandle, 1u);
}

/* [] END OF FILE */

```